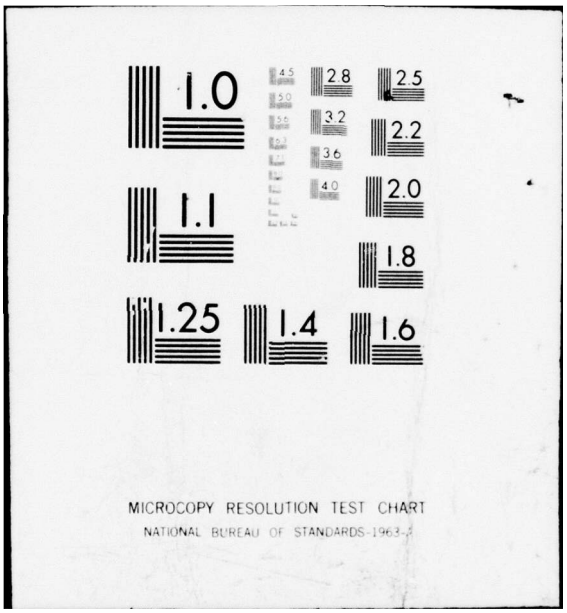
AD-A038 053 AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO F/G 9/2
EFFICIENCY COMPARISON OF JOVIAL-73/I AND AN/AYK-15 ASSEMBLY LAN--ETC(U)
JAN 77 W L TRAINOR, M BURLAKOFF, J GARRETT

UNCLASSIFIED AFAL-TR-76-253 NL

|OF|
AD
A038053

END

DATE
FILMED
4-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFAL-TR-76-253

12

EFFICIENCY COMPARISON OF JOVIAL-73/I AND AN/AYK-15 ASSEMBLY LANGUAGE

DAIS PROJECT OFFICE
SYSTEM AVIONICS DIVISION

JANUARY 1977

TECHNICAL REPORT AFAL-TR-76-253
FINAL REPORT FOR PERIOD AUGUST - SEPTEMBER 1976

Approved for public release; distribution unlimited

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

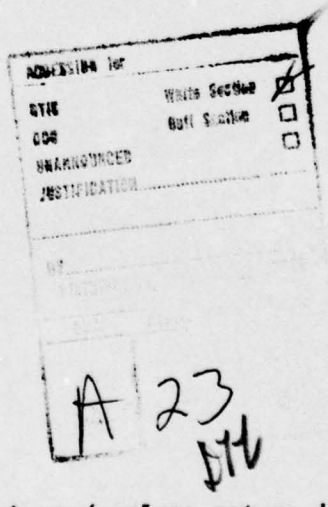
When government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto. This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NITS). At NITS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

W. LYNN TRAINOR
Technical Manager,
DAIS Software Group

FOR THE COMMANDER

JAMES D. EVERETT, Col, USAF
Chief, System Avionics Division
AF Avionics Laboratory



A 23

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE - 4 APRIL 77 - 100

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFAL-TR-76-253	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: Efficiency Comparison of JOVIAL-73/I and AN/AYK-15 Assembly Language	5. TYPE OF REPORT & PERIOD COVERED: Final Report, Aug – Sep 1976
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Mssrs. W. Lynn Trainor, Mike Burlakoff, and John Garrett	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Avionics Laboratory (AFAL/AA) Wright-Patterson AFB, Oh 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Proj/Tsk 2052/02
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Avionics Laboratory (AFAL/AA) Wright-Patterson AFB, Oh 45433	12. REPORT DATE: Jan [redacted]77
	13. NUMBER OF PAGES: 86
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Digital Avionics Information System
Avionics Software
JOVIAL
Higher Order Language

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report documents and discusses the results of a project undertaken to assess the object code efficiency of the Digital Avionics Information System (DAIS) JOVIAL-73/I compiler relative to manual coding in assembly language. The DAIS JOVIAL-73/I compiler was designed to produce highly efficient object code in terms of both memory size requirements and program execution time requirements. Two sample algorithms were coded in both JOVIAL-73/I and assembly language for the AN/AYK-15 airborne computer. The compiler-produced object code was, in turn, compared to the manually produced assembly language versions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

011 670

FOREWORD

At the request of the Joint Tactical Information Distribution System (JTIDS), Joint Program Office (JPO), the Air Force Avionics Laboratory undertook this programming language comparison effort. The overall objective was to compare the relative efficiencies of JOVIAL-73/I and Assembly Language coding for the AN/AYK-15 airborne computer. This effort was conducted in-house by the Digital Avionics Information System (DAIS) Project Office of the System Avionics Division. The algorithms were coded by Mr. John Garrett, and the analysis and report writing were performed by Mr. William L. Trainor and Mr. Mike Burlakoff.

TABLE OF CONTENTS

SECTION I

INTRODUCTION

This report was generated to document the results of an in-house study effort conducted by the Digital Avionics Information System (DAIS) Project Office, System Avionics Division, Air Force Avionics Laboratory. This effort was undertaken to support the JPO/MITRE 1976 Summer Study which is attempting to define a data processing strategy for the Joint Tactical Information Distribution System (JTIDS) project. The objective of the DAIS support effort was to obtain and analyze language efficiency data on the usage of the JOVIAL-73/I language as compared to assembly language programming. In particular, two algorithms (see appendices "A" and "B") were chosen that were considered to be typical of the computational and data extraction activities central to the JTIDS data processing environment. In turn these two algorithms were coded in both JOVIAL-73/I and assembler language for the AN/AYK-15 airborne computer, and information was obtained on the programmer's coding time, the computer storage, and the execution time requirements.

As a part of the overall JPO/MITRE Study, the data of this report will be used as an input to defining the JTIDS data processing strategy. Similar coding comparison efforts are concurrently underway implementing the same algorithms in COBOL, FORTRAN, and the JTIDS Standard Instruction Set.

SECTION II

COMPARISON PROCEDURES

PROCEDURES

The two JTIDS algorithms were each coded in both JOVIAL-73/I and assembler language, and in turn, the four resultant programs were compiled for the AN/AYK-15 airborne computer. For this effort, the following salient features are notable:

1. The coding was performed by an experienced programmer, following the normal DAIS program production processes. Only one programmer was involved, and he performed both the JOVIAL-73/I and assembly language coding. This individual was slightly more proficient with the JOVIAL-73/I language than the AN/AYK-15 Assembly Language, however this difference was considered very minimal.

2. The programs were developed to the point that error-free compilations were available, and no attempts were made to debug the actual logic of the routines. Note that the "execution time" data obtained was by a manual process of adding instruction times and not by actual program executions.

3. A log was kept, to the nearest half-hour, of programmer time spent on each algorithm. This time includes only that required to code the particular algorithms and correct them to the point of obtaining error-free compilations. It does not include the time required for data analysis and report writing.

SECTION II

COMPARISON PROCEDURES

PROCEDURES

The two JTIDS algorithms were each coded in both JOVIAL-73/I and assembler language, and in turn, the four resultant programs were compiled for the AN/AYK-15 airborne computer. For this effort, the following salient features are notable:

1. The coding was performed by an experienced programmer, following the normal DAIS program production processes. Only one programmer was involved, and he performed both the JOVIAL-73/I and assembly language coding. This individual was slightly more proficient with the JOVIAL-73/I language than the AN/AYK-15 Assembly Language, however this difference was considered very minimal.

2. The programs were developed to the point that error-free compilations were available, and no attempts were made to debug the actual logic of the routines. Note that the "execution time" data obtained was by a manual process of adding instruction times and not by actual program executions.

3. A log was kept, to the nearest half-hour, of programmer time spent on each algorithm. This time includes only that required to code the particular algorithms and correct them to the point of obtaining error-free compilations. It does not include the time required for data analysis and report writing.

4. The storage and run-time requirements for the sine/cosine procedures were not included in the data. However, the calling-conventions overhead is included as part of the data.

5. The algorithm development efforts were performed in an interactive, time-shared environment using the DEC-10 computer system. The programmer was responsible for coding the algorithm, entering the source lines via a time-share terminal using a text editor, and compiling these programs via the same time-share facilities.

In order to make the resultant data as directly comparable as possible, several coding "ground rules" were followed for both the assembly language efforts and JOVIAL-73/I efforts. These were:

1. A standard procedure-linkage convention was selected and was used for both the JOVIAL-73/I and assembly language implementations.

2. The flowcharts and algorithm information supplied by the JTIDS office was followed exactly for both the JOVIAL-73/I and assembly language implementations. One variation was taken with the "ACCEPT/HASH/STORE" algorithm which resulted in a restructuring of the logic (see Appendix "C"), but this program (MBLT1) was separately documented.

3. No explicit attempts were made to optimize the coding other than the use of "normal" coding practices.

ALGORITHMS: Two algorithms were implemented as these were supplied by the JTIDS Project Office. Each is briefly discussed below. A more complete description is given in Appendices "A" and "B", and those descriptions are in essence the level of information given to the programmer.

1. Algorithm #1, Coordinate Conversion: This algorithm is a very straight-forward, mathematically oriented algorithm using mainly algebraic and trigonometric relationships. Little or no "logic" is required for this code. The algorithm is designed to convert from the latitude and longitude reference frame of JTIDS messages to the flat coordinate reference frame used by SAGE, E-3A, or 485L. Appendix "A" contains the detailed description of this algorithm along with the resultant JOVIAL-73/I code (procedure name EQUA) and assembly language code (procedure name EQUAMB).

2. Algorithm #2, ACCEPT/HASH/STORE: Contrary to the coordinate conversion problem above, the second algorithm is almost entirely "logic" and little or no mathematical computation. The algorithm examines a received JTIDS message for the LIBRARY or INPUT MESSAGE MANAGEMENT function, and codes bits to indicate whether the message uses simulated or live data and whether it contained friendly or hostile ground, air or sea data, etc. This algorithm was implemented in two ways:

a. The flow charts were first implemented with logic exactly as received from the JTIDS office. These results are contained in Appendix "B". The procedure ACHAST is the assembly language version, and the procedure MBLT is the JOVIAL-73/I version.

b. The flow chart was next modified to implement a more "structured" algorithm per the guidelines of structured programming. The resulting flow chart and listing (JOVIAL-73/I only) is contained in Appendix "C". This procedure is entitled MBLT1.

SECTION III

RESULTS AND CONCLUSIONS

Results: Table 1 is a compilation of the results obtained. The six rows of this table indicate the following:

a. Rows one and two are the assembly language and JOVIAL-73/I versions, respectively, of Algorithm #1, Coordinate Conversion. The "execution time" data is a result of a manual addition of AN/AYK-15 instruction times for each object code instruction produced by the language translators.

b. Rows three and four are the assembly language and JOVIAL-73/I versions, respectively, of Algorithm #2, ACCEPT/HASH/STORE. Representative execution time data could not be obtained for this algorithm due to the large number of possible control paths present in the algorithm.

c. Row five is a JOVIAL-73/I implementation of a "structured programming" version of Algorithm #2, ACCEPT/HASH/STORE.

d. Row six is the relevant data (programmer time) required for construction of the needed COMPOOL. This COMPOOL is used with both the assembly language and JOVIAL-73/I implementations, and the COMPOOL time should be added to the times above to obtain total representative time estimates when "starting from scratch."

Conclusions: The following major conclusions and observations apply.

1. An approximate 10% inefficiency is incurred with the JOVIAL-73/I implementations. These results compare favorably with other results published for other comparable HOL's that have indicated a 10% to 20% range. This inefficiency appears to hold both for memory usage and execution time.

TABLE 1

SUMMARY RESULTS

ALGORITHM	FILE NAME	LANGUAGE/ METHOD	MEMORY (16-BIT WORDS		EXECUTION TIME		PROGRAMMER TIME	
			NUMBER OF WORDS	% EXPANSION OVER ASSEMBLY LANGUAGE	MICRO-SEC	% EXPANSION OVER ASSEMBLY LANGUAGE	CODING	KEY PUNCH & COMPILE
ALGORITHM #1, "COORDINATE CONVERSION"	EQUAMB	Westinghouse Assembler	279	-	316.	-	6.0	2,5
	EQUA	J73/I	308	10.4%	347.4	9.8%	2.5	1.0
ALGORITHM #2, "ACCEPT/ HASH/ STORE"	ACHAST	Westinghouse Assembler (Original Flowchart)	429	-	-	-	23.	4.
	MBLT	J73/I (Original Flowchart)	480	11.9%	-	-	10.	2.5
	MBLT1	J73/I (Structured Logic and Structured Flowchart)	538	25.4%	-	-	7.5	3.0
J73/I COMPOOL (Used for Algorithms)	JTIDS	J73/I	0	-	-	-	.5	.25

2. The programmer's productivity is markedly better in JOVIAL-73/I; better than two to one.

3. The JOVIAL-73/I implementations are much easier to read and interpret than the assembly language versions. The reliability, maintainability, and "modifiability" should likewise be much better for the JOVIAL-73/I versions.

4. With more programmer time allotted, both the JOVIAL-73/I and assembly language versions could be improved, efficiency wise.

5. The major inefficiency with the JOVIAL-73/I versions appears to be in sub-optimal usage of the available registers. With an improved optimizer algorithm in the compiler, this 10% figure could be significantly reduced. Study efforts are currently underway to improve this optimizer.

6. A very general procedure linkage convention was used for both the assembly language and JOVIAL-73/I versions of each algorithm. A more efficient convention is presently being defined for use in the DAIS Project, and this convention should significantly reduce the size and execution time of both assembly language and JOVIAL-73/I programs.

7. The "structured" version of Algorithm #2, MBLT1, is much more readable and understandable than the original version, MBLT. However, the cost was an approximate 14% further inefficiency for this particular implementation. It is felt that with sufficient information on the ACCEPT/HASH/STORE algorithm, a complete redesign could be accomplished with structured programming principles that would be as efficient as the "unstructured" JOVIAL case (MBLT). In addition to HOL alternatives for JTIDS, it is felt that some consideration should be given to "structured control" alternatives

(e.g., the MBLT1 example) since these produce much superior algorithms from a "maintainability" standpoint.

APPENDIX A

ALGORITHM #1, COORDINATE CONVERSION

ALGORITHM #1 (COORDINATE CONVERSION) DESCRIPTION:

The equations convert from the latitude and longitude reference frame messages to the flat coordinate plane used for SAGE, E-3A or 485L.

All symbols are defined below:

Definitions

X,Y = track position relative to sector center, nautical miles

$Lp,\lambda p$ = latitude and longitude of sector center

L,λ = reported latitude and longitude of track position

$\Delta L,\Delta\lambda$ = differential latitude and longitude minutes

Ep = earth radius at sector center, nautical miles

ϕ = conformal latitude

θ = reported heading relative to true north, 0-359 degrees

S = reported speed, 0-2047 data miles per hour

$\dot{X},\dot{Y}$ = components of track velocity, knots

Conversion equations

Find $\dot{X},\dot{Y}$ as follows:

$$\dot{X} = .987475\ S \sin\theta$$

$$\dot{Y} = .987475\ S \cos\theta$$

Note: All position data is positive east and north, negative west and south except differential longitude which is positive west and negative east.

Find, X,Y as follows:

$$X = \frac{2E_p \sin\Delta\lambda_p \cos\phi}{1 + \sin\phi \sin\phi_p + \cos\phi \cos\phi_p \cos\Delta\lambda_p}$$

$$Y = \frac{2E_p(\sin\phi\cos\phi_p - \cos\phi\sin\phi_p\cos\Delta\lambda_p)}{1 + \sin\phi\sin\phi_p + \cos\phi\cos\phi_p\cos\Delta\lambda_p}$$

Find sin ϕ, cos ϕ and sin ϕ_p, cos ϕ_p as follows (substituting L_p for L when finding functions of ϕ_p):

$$\text{Sin}\ \phi = \sin L\ (0.99327733 + 0.00666251\ \sin^2 L + 0.00005959\ \sin^4 L + 0.00000059\ \sin^6 L)$$

$$\text{or sin}\ \phi = \tan^{-1}(0.99327733\ \tan L)$$

$$\cos\phi = (1 - \sin^2\phi)^{\frac{1}{2}}$$

Find E_p as follows:

$$E_p = \frac{3444.054\ \cos L_p}{\cos\phi_p\ (1 - 0.00672267\ \sin L_p)^{\frac{1}{2}}}$$

ASSEMBLY LANGUAGE IMPLEMENTATION

The following pages are the resultant assembler output for Algorithm #1, Coordinate Conversion. The procedure name is EQUAMB.

```
EEEEEEEEEEEEEEE      QQQQQQQQQ      UUU     UUU      AAAAAAAAA      MMM       MMM      BBBBBBBBBBBBB
EEEEEEEEEEEEEEE      QQQQQQQQQ      UUU     UUU      AAAAAAAAA      MMM       MMM      BBBBBBBBBBBBBB
EEEEEEEEEEEEEEE      QQQQQQQQQ      UUU     UUU      AAAAAAAAA      MMM       MMM      BBBBBBBBBBBBBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMMMMMM   MMMMMMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMMMMMM   MMMMMMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMMMMMM   MMMMMMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMM MMM MMM MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMM MMM MMM MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMM MMM MMM MMM     BBB     BBB
EEEEEEEEEEE        QQQ     QQQ     UUU     UUU     AAA     AAA     MMM       MMM       BBBBBBBBBBBB
EEEEEEEEEEE        QQQ     QQQ     UUU     UUU     AAA     AAA     MMM       MMM       BBBBBBBBBBBB
EEEEEEEEEEE        QQQ     QQQ     UUU     UUU     AAA     AAA     MMM       MMM       BBBBBBBBBBBB
EEE                QQQ     QQQ     UUU     UUU     AAAAAAAAAAAAAAA     MMM       MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAAAAAAAAAAAAAA     MMM       MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAAAAAAAAAAAAAA     MMM       MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMM       MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMM       MMM     BBB     BBB
EEE                QQQ     QQQ     UUU     UUU     AAA     AAA     MMM       MMM     BBB     BBB
EEEEEEEEEEEEEEE      QQQQQQ    QQQ     UUUUUUUUUUUUUUU     AAA     AAA     MMM       MMM     BBBBBBBBBBBBB
EEEEEEEEEEEEEEE      QQQQQQ    QQQ     UUUUUUUUUUUUUUU     AAA     AAA     MMM       MMM     BBBBBBBBBBBBBB
EEEEEEEEEEEEEEE      QQQQQQ    QQQ     UUUUUUUUUUUUUUU     AAA     AAA     MMM       MMM     BBBBBBBBBBBBBB

LLL                SSSSSSSSSSS     TTTTTTTTTTTTTTT
LLL                SSSSSSSSSSS     TTTTTTTTTTTTTTT
LLL                SSSSSSSSSSS     TTTTTTTTTTTTTTT
LLL              SSS               TTT
LLL              SSS               TTT
LLL              SSS               TTT
LLL              SSS               TTT
LLL              SSS               TTT
LLL              SSS               TTT
LLL                SSSSSSSSS       TTT
LLL                SSSSSSSSS       TTT
LLL                SSSSSSSSS       TTT
LLL                        SSS     TTT
LLL                        SSS     TTT
LLL                        SSS     TTT
LLL                        SSS     TTT
LLL                        SSS     TTT
LLL                        SSS     TTT
LLLLLLLLLLLLLLL    SSSSSSSSSSS     TTT
LLLLLLLLLLLLLLL    SSSSSSSSSSS     TTT
LLLLLLLLLLLLLLL    SSSSSSSSSSS     TTT

LPTSPL Version 6(347)  Running on LPTV11
 *START* User BURLAKOFF [3242,1376] Job  JTIDS Seq. 521 Date 14-Sep-76 10:03:39 Monitor AFAL 602.15 SYSTEM *START*
Request created: 14-Sep-76  10:04:11
File: DSK01:EQUAMB.LST[3242,1376] Created: 14-Sep-76 10:00:00 Printed: 14-Sep-76 10:22:39
QUEUE Switches:  /PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:305 /FORMS:NORMAL
File will be deleted after printing
```

```
                *****MEMORY ALLOCATION TABLE FOR  COORTR   ****

 REFERENCE      DECIMAL MEMORY LOCATIONS          EQUIVALENT HEX MEM LOCATIONS
   INDEX        FROM        TO        TYPE        FROM        TO        TYPE
     1           0          45         S          0000        002D       S
     2          46          67         D          002E        0043       D
     3          68         279         C          0044        0117       C
1    PROGRAM;  COORTR     PAGE   1      DAIS HBC ASSEMBLER; VERSION 001

 LINE   LOCATION      HEX CODE A   MANUSCRIPT ,

    1                                        MODULE  COORTR
    2                                 .
    3                                 .      COORDINATE TRANSFORM
    4                                 .
    5                                 .      REGISTER DEFINES
    6                                 .
    7                                 R0     EQU     0
    8                                 R1     EQU     1
    9                                 R2     EQU     2
   10                                 R3     EQU     3
   11                                 R4     EQU     4
   12                                 R5     EQU     5
   13                                 R6     EQU     6
   14                                 R7     EQU     7
   15                                 R8     EQU     8
   16                                 R9     EQU     9
   17                                 R10    EQU     10
   18                                 R11    EQU     11
   19                                 R12    EQU     12
   20                                 R13    EQU     13
   21                                 R14    EQU     14
   22                                 R15    EQU     15
   23                                 .
   24                                 .      INTERNAL DECLARATIONS
   25                                 .
   26                                        ENTRY   COORTR
   27                                 .
   28                                 .      EXTERNAL DECLARATIONS
   29                                 .
   30                                        EXTERNAL        SIN
   31                                        EXTERNAL        COS
   32                                        EXTERNAL        ATAN
   33                                        EXTERNAL        TAN
   34                                        EXTERNAL        SQRT
   35                                 .
   36                                 .      CONDITION DEFINES
   37                                 .
   38                                 LT     EQU     1
   39                                 EQ     EQU     2
   40                                 LE     EQU     3
   41                                 GT     EQU     4
   42                                 NE     EQU     5
   43                                 GE     EQU     6
```

```
                                                .
                                                .       LOCAL DATA ALLOCATION
                                                .
0000            0000            XX      STORAGE 2
0002            0000            YY      STORAGE 2
0004            0000            XX'     STORAGE 2
0006            0000            YY'     STORAGE 2
0008            0000            LONG'CEN        STORAGE 2
000A            0000            LAT'CENT        STORAGE 2
000C            0000            LAT'TRAC        STORAGE 2
000E            0000            EARTH'RA        STORAGE 2
1   PROGRAM;  COORTR    PAGE    2    DAIS HHC ASSEMBLER; VERSION 001

 LINE  LOCATION      HEX CODE A  MANUSCRIPT  ,

0010            0000            COS'CENT        STORAGE 2
0012            0000            COS'TRAC        STORAGE 2
0014            0000            SIN'CENT        STORAGE 2
0016            0000            SIN'TRAC        STORAGE 2
0018            0000            HEADING STORAGE 2
001A            0000            SPEED   STORAGE 2
001C            0000            DIVISOR STORAGE 2
001E            0000            SAVE    STORAGE 10
0028            0000            TMP     STORAGE 2
002A            0000            TMP1    STORAGE 2
002C            0000            ANS     STORAGE 2
002E    7E65    9400            CXX     CONSTANT        07E65,09400
0030    7F23    B500            CSIN    CONSTANT        07F23,0B500
0032    6BA0    6E0C            CERAD   CONSTANT        06BA0,06E0C
0034    4000    0001            CONE    CONSTANT        04000,00001
0036    4000    0001            CTWO    CONSTANT        04000,00001
0038    6E24    EBF9            CERAD2  CONSTANT        06E24,0EBF9
003A            0018            PLIST   CONSTANT        HEADING
003B            002C                    CONSTANT        ANS
003C            000C            PLIST1  CONSTANT        LAT'TRAC
003D            002C                    CONSTANT        ANS
003E            0028            PLIST2  CONSTANT        TMP
003F            002C                    CONSTANT        ANS
0040            000A            PLIST3  CONSTANT        LAT'CENT
0041            002C                    CONSTANT        ANS
0042            0008            PLIST4  CONSTANT        LONG'CEN
0043            002C                    CONSTANT        ANS
                                                .
                                                .       ENTRY TO PROGRAM COORTR
                                                .
                                                .       SAVE REGISTERS
0044    9F90    001E    R       COORTR  STM     R9,SAVE
                                        .       FIND X VELOCITY COMPONENT
                                        .       XX'=.987475 * SPEED *SIN(HEADING)
                                        .
                                        .       GET SIN OF HEADING
0046    83F0    003A    R               LIM     R15,PLIST
0048    72ZF    FFFF    X               JS      R2,SIN
                                        .       MULTIPLY SIN(HEADING)*SPEED*CXX
004A    8400    002E    R               DL      R0,CXX
004C    C700    001A    R               FM      R0,SPEED
004E            8520    A               CLR     R2,R0
004F            FF00    A               NOP
0050    C720    002C    R               FM      R2,ANS
```

```
    99                                          .       STORE   XX'
   100      0052   9420   0004   R                    DST     R2,XX'
   101                                          .       FIND Y VELOCITY COMPONENT
   102                                          .       YY' = .987475 * SPEED *COS(HEADING)
   103                                          .
   104      0054   9400   0028   R                    DST     0,TMP
   105      0056   83F0   003A   R                    LIM     R15,PLIST
   106      0058   7220   FFFF   X                    JS      R2,COS
   107      005A   8400   0028   R                    DL      0,TMP
   108      005C   C700   002C   R                    FM      R0,ANS
1    PROGRAM:  COORTR      PAGE    3     DAIS HBC ASSEMBLER; VERSION 001

  LINE   LOCATION        HEX CODE A    MANUSCRIPT


   109                                          .       STORE IN YY'
   110      005E   9400   0006   R                    DST     R0,YY'
   111                                          .       FIND SIN AND COS OF TWO COORDINATE FRAMES
   112                                          .
   113                                          .       SIN'TRACK=ATAN(.99327733 * TAN(LAT'TRACK))
   114                                          .
   115                                          .       FIND TAN(LAT'TRAC)
   116      0060   83F0   003A   R                    LIM     15,PLIST
   117      0062   7220   FFFF   X                    JS      R2,TAN
   118                                          .       MULTIPLY TAN(LAT'TRAC)*CSIN
   119      0064   8400   0030   R                    DL      R0,CSIN
   120      0066   C700   002C   R                    FM      R0,ANS
   121      0068   9400   0028   R                    DST     R0,TMP
   122      006A   83F0   003C   R                    LIM     R15,PLIST1
   123                                          .       TAKE ATAN OF VALUE
   124      006C   7220   FFFF   X                    JS      R2,ATAN
   125      006E   8400   002C   R                    DL      R0,ANS
   126                                          .       STORE IN SIN'TRAC
   127      0070   9400   0016   R                    DST     R0,SIN'TRAC
   128                                          .       SIN'CENTER=ATAN(.99327733 * TAN(LAT'CENTER))
   129                                          .
   130                                          .       MULTIPLY TAN(LAT'CENT) BY CSIN
   131      0072   83F0   003E   R                    LIM     R15,PLIST2
   132      0074   7220   0063   X                    JS      R2,TAN
   133      0076   8400   0030   R                    DL      R0,CSIN
   134      0078   C700   002C   R                    FM      R0,ANS
   135      007A   9400   0028   R                    DST     R0,TMP
   136      007C   83F0   003E   R                    LIM     R15,PLIST2
   137                                          .       TAKE ATAN OF VALUE
   138      007E   7220   006D   X                    JS      R2,ATAN
   139      0080   8400   002C   R                    DL      R0,ANS
   140                                          .       STORE IN SIN'CENT
   141      0082   9400   0014   R                    DST     R0,SIN'CENT
   142                                          .       FIND COS'TRAC
   143                                          .       COS'TRACK = SQRT(1 - (SIN'TRACK)**2.)
   144                                          .
   145      0084   8420   0016   R                    DL      R2,SIN'TRAC
   146      0086   C720   0016   R                    FM      R2,SIN'TRAC
   147      0088   9420   0028   R                    DST     R2,TMP
   148      008A   8440   0034   R                    DL      R4,CONE
   149      008C   8740   0028   R                    FS      R4,TMP
   150      008E   9440   0028   R                    DST     R4,TMP
   151      0090   83F0   003E   R                    LIM     R15,PLIST2
   152      0092   7220   FFFF   X                    JS      R2,SQRT
   153      0094   8400   002C   R                    DL      R0,ANS
```

```
 154                                        .       STORE IN COS'TRAC
 155      0096  9420  0012  R                       DST     R0,COS'TRAC
 156                                        .       FIND COS'CENT
 157                                        .       COS'CENTER = SQRT(1 - (SIN'CENTER)**2.)
 158                                        .
 159      0098  8420  0014  R                       DL      R2,SIN'CENT
 160      009A  C720  0014  R                       FM      R2,SIN'CENT
 161      009C  9420  0028  R                       DST     R2,TMP
 162      009E  8440  0034  R                       DL      R4,CONE
 163      00A0  B740  0028  R                       FS      R4,TMP
1    PROGRAM:  COORTR      PAGE    4     DAIS HBC ASSEMBLER; VERSION 001

  LINE   LOCATION        HEX CODE A    MANUSCRIPT

 164      00A2  9440  0028  R                       DST     R4,TMP
 165      00A4  83F0  003E  F                       LIM     R15,PLIST2
 166      00A6  7220  0093  X                       JS      R2,SQRT
 167      00A8  8420  002C  R                       DL      R2,ANS
 168                                        .       STORE IN COS'CENT
 169      00AA  9420  0010  R                       DST     R2,COS'CENT
 170                                        .       FIND EARTH RADIUS
 171                                        .       EARTH'RADIUS = (3444.054 * COS(LAT'CENTER))/(COS'CENTER*SQRT(1 - .006722
 172                                        .
 173      00AC  83F0  0040  R                       LIM     R15,PLIST3
 174      00AE  7220  0049  X                       JS      R2,SIN
 175      00B0  8400  002C  R                       DL      R0,ANS
 176      00B2  C700  002C  R                       FM      R0,ANS
 177      00B4  C700  0038  R                       FM      R0,CERAD2
 178      00B6  9400  0028  R                       DST     R0,TMP
 179      00B8  8420  0034  R                       DL      R2,CONE
 180      00BA  B720  0028  R                       FS      R2,TMP
 181      00BC  83F0  003E  R                       LIM     R15,PLIST2
 182      00BE  7220  00A7  X                       JS      R2,SQRT
 183      00C0  8420  002C  R                       DL      R2,ANS
 184      00C2  C720  0010  R                       FM      R2,COS'CENT
 185      00C4  9420  0028  R                       DST     R2,TMP
 186      00C6  83F0  0040  R                       LIM     R15,PLIST3
 187      00C8  7220  0059  X                       JS      R2,COS
 188      00CA  8420  002C  R                       DL      R2,ANS
 189      00CC  C720  0032  R                       FM      R2,CERAD
 190      00CE  D720  0028  R                       FD      R2,TMP
 191                                        .       STORE EARTH RADIUS
 192      00D0  9420  000E  R                       DST     R2,EARTH'RA
 193                                        .       OBTAIN COMMON DIVISOR
 194                                        .       DIVISOR = 1 + SIN'CENTER * SIN'TRACK + COS'CENTER*COS'TRACK*COS(LONG'CEN
 195                                        .
 196      00D2  83F0  0042  R                       LIM     R15,PLIST4
 197      00D4  7220  00C9  X                       JS      R2,COS
 198                                        .       FIND COS'CENT*COS'TRAC*COS(LONG'CEN)
 199      00D6  8400  002C  R                       DL      R0,ANS
 200      00D8  C700  0012  R                       FM      R0,COS'TRAC
 201      00DA  C700  0010  R                       FM      R0,COS'CENT
 202      00DC  9400  0028  R                       DST     R0,TMP
 203                                        .       FIND    SIN'CENT*SIN'TRAC
 204      00DE  8400  0014  R                       DL      R0,SIN'CENT
 205      00E0  C700  0016  R                       FM      R0,SIN'THAC
 206                                        .       FIND TMP+R0
 207      00E2  A700  0028  R                       FA      R0,TMP
 208      00E4  A700  0034  R                       FA      R0,CONE
```

```
209                                    .       STORE IN DIVISOR
210     00E6  9400  001C   R                  DST     R0,DIVISOR
211                                    .       FIND X POSITION
212                                    .       XX = (2.*EARTH'RADIUS*SIN(LONG'CENTER)*COS'TRACK)/DIVISOR
213                                    .
214                                    .       FIRST   2.*EARTH'RA
215     00E8  8400  003B   R                  DL      R0,CTWO
216     00EA  C700  0006   R                  FM      R0,EARTH'RA
217     00EC  9400  002A   R                  DST     R0,TMP
1    PROGRAM;  COORIK      PAGE    5     DAIS HBC ASSEMBLER; VERSION 001

 LINE   LOCATION      HEX CODE A   MANUSCRIPT

218     00EE  C700  0012   R                  FM      R0,COS'TRAC
219     00F0  9400  002A   R                  DST     R0,TMP1
220                                    .       FIND SIN(LONG'CEN)
221     00F2  83E0  0042   R                  LIM     R15,PLIST4
222     00F4  7220  00AF   X                  JS      R2,SIN
223                                    .       NOW TMP1*SINA/DIVISOR
224     00F6  8400  002A   R                  DL      R0,TMP1
225     00F8  C700  002C   R                  FM      R0,ANS
226     00FA  D700  001C   R                  FD      R0,DIVISOR
227                                    .       STORE X POSITION
228     00FC  9400  0000   R                  DST     R0,XX
229                                    .       FIND Y POSITION
230                                    .       YY = (2.*EARTH'RADIUS*(SIN'TRACK*COS'CENTER - COS'TRACK*SIN'CENTER*COS(L
231                                    .
232     00FE  83E0  0042   R                  LIM     R15,PLIST4
233     0100  7220  0005   X                  JS      R2,COS
234                                    .       TMP1*SIN'CENT*COS(LONG'CEN)
235     0102  8400  002A   R                  DL      R0,TMP1
236     0104  C700  0014   R                  FM      R0,SIN'CENT
237     0106  C700  002C   R                  FM      R0,ANS
238     0108  9400  002A   R                  DST     R0,TMP1
239                                    .       TMP*SIN'TRAC*COS'CENT
240     010A  8400  002A   R                  DL      R0,TMP
241     010C  C700  0016   R                  FM      R0,SIN'TRAC
242     010E  C700  0010   R                  FM      R0,COS'CENT
243                                    .       SUBTRACT TMP1
244     0110  8700  002A   R                  FS      R0,TMP1
245                                    .       STORE   YY POSITION
246     0112  9400  0002   R                  DST     R0,YY
247     0114  8F90  001F   R          OUT     LM      R9,SAVE
248     0116  7DF2  0000   A                  J       0,R2
249                                            END

  *  NO START ADDRESS ENCOUNTERED

   0  ERROR(S) THIS RUN.


1 SYMBOLS              ** ASSIGNMENT **      LINE       MUL DEF       ATTRIBUTE
                       HEX          DECIMAL
ANS                    002C           44        65                      R

ATAN                   007F          127        32                      EX

CERAD                  0032           50        68                      R
```

CFRAD2	0038	56	71		R
CONE	0034	52	69		R
COORTP	0044	68	86		RE
COS	0101	257	31		EX
COS'CENT	0010	16	55		R
COS'TRAC	0012	18	56		R
CSIN	0030	48	67		R
CTWO	0036	54	70		R
CXX	002E	46	66		R
DIVISOR	001C	28	61		R
EARTH'RA	000E	14	54		R
EQ	0002	2	39		A
GE	0006	6	43		A
GT	0004	4	41		A
HEADING	0018	24	59		R
LAT'CENT	000A	10	52		R
LAT'TRAC	000C	12	53		R
LE	0003	3	40		A
LONG'CEN	0008	8	51		R
LT	0001	1	38		A
NE	0005	5	42		A
OUT	0114	276	247		R
PLIST2	003E	62	76		R
PLIST3	0040	64	78		R
I SYMBOLS	** ASSIGNMENT ** HEX	DECIMAL	LINE	MUL DEF	ATTRIBUTE
PLIST4	0042	66	80		R
PLIST	003A	58	72		R
PLIST1	003C	60	74		R
R0	0000	0	7		A
R1	0001	1	8		A

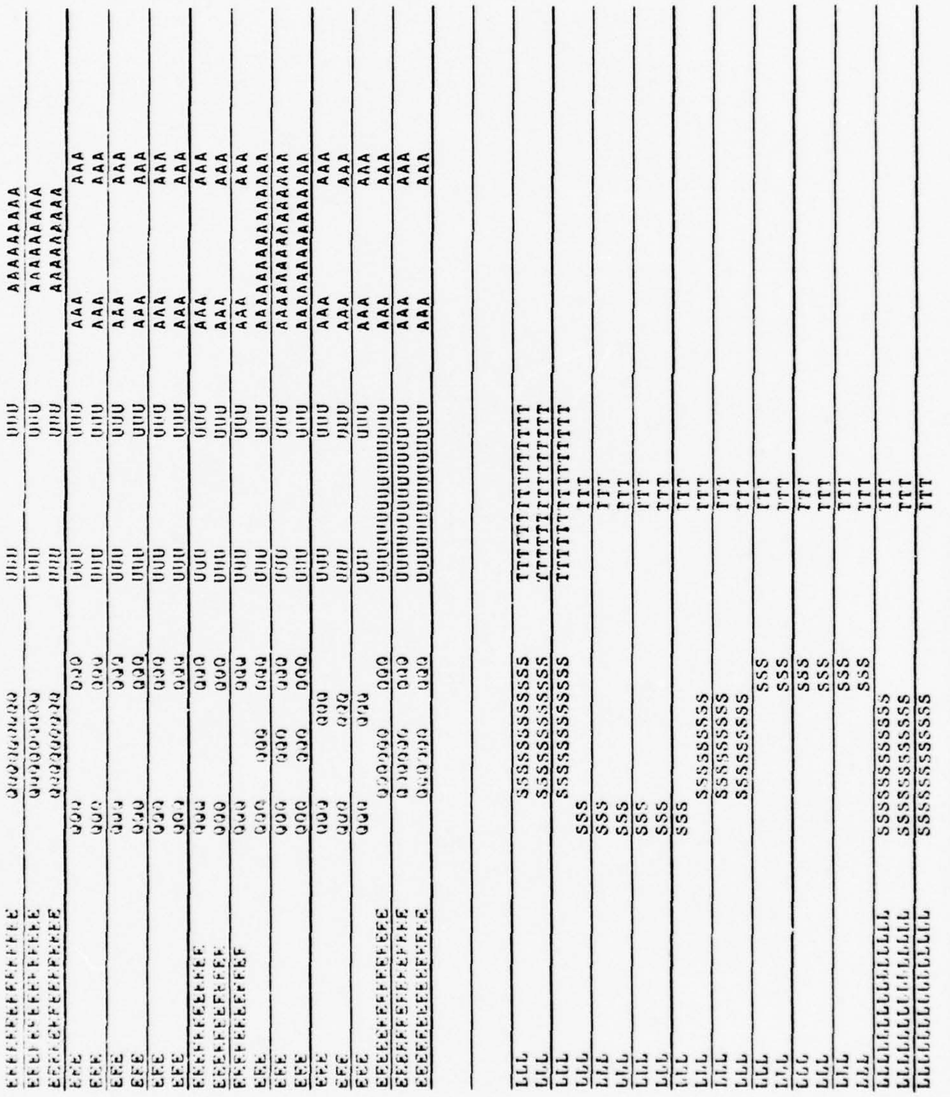
R10	000A	10	17	A
R11	000B	11	18	A
R12	000C	12	19	A
R13	000D	13	20	A
R14	000E	14	21	A
R15	000F	15	22	A
R2	0002	2	9	A
R3	0003	3	10	A
R4	0004	4	11	A
R5	0005	5	12	A
R6	0006	6	13	A
R7	0007	7	14	A
R8	0008	8	15	A
R9	0009	9	16	A
SAVE	001E	30	62	R
SIN	00F5	245	30	EX
SIN'CENT	0014	20	57	R
SIN'FRAC	0016	22	58	R
SPEED	001A	26	60	R
SQRT	00BF	191	34	EX
TAN	0075	117	33	EX
TMP	0028	40	63	R

I SYMBOLS	** ASSIGNMENT ** HEX	DECIMAL	LINE	MUL DEF	ATTRIBUTE
TMP1	002A	42	64		R
XX	0000	0	47		R
XX'	0004	4	49		R
YY	0002	2	48		R
YY'	0006	6	50		R

ELAPSED TIME = 65 SECONDS

JOVIAL-73/I IMPLEMENTATION

The following pages are the resultant compiler output for the JOVIAL-73/I compilation of Algorithm #1, Coordinate Conversion. The procedure name is EQUA.




```
LPTSPL version 6(347)   Running on LPT011
*START* User MERLAKOFF [32V2,1376] Job JTIDS Seq. 521 Date 14-Sep-76 10:03:39 Monitor AFAL 602.15 SYSTEM *START*
Request created: 14-Sep-76 10:04:11
File: DSKD1:EQUA.LST[32V2,1376] Created: 14-Sep-76 09:44:00 Printed: 14-Sep-76 10:21:09
QUEUE Switches: /PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:324 /FORMS:NORMAL
File will be deleted after printing
```

```
EQUA,EQUA=EQUA.JTS/HBC/MAC/ACR/STAT/NOIN/NOPT

    1.          !COMPOOL('JTIDS.CMP');
    2.          PROC   COORDINATE'TRANSFORMATION;
    3.                 BEGIN   "COORDINATE'TRANSFORMATION"
    3.
    3.                         "DECLARATION SECTION"
    3.                         "DATA DECLARATION"
    3.                 ITEM    XX      F;
    4.                 ITEM    YY      F;
    5.                 ITEM    XX'     F;
    6.                 ITEM    YY'     F;
    7.                 ITEM    LONG'CENTER     F;
    8.                 ITEM    LAT'CENTER      F;
    9.                 ITEM    LAT'TRACK       F;
   10.                 ITEM    EARTH'RADIUS    F;
   11.                 ITEM    COS'PHI'CENTER  F;
   12.                 ITEM    COS'PHI'TRACK   F;
   13.                 ITEM    SIN'PHI'CENTER  F;
   14.                 ITEM    SIN'PHI'TRACK   F;
   15.                 ITEM    HEADING F;
   16.                 ITEM    SPEED   F;
   17.                 ITEM    DIVISOR F;
   18.
   18.                         "FIND VELOCITY COMPONENTS"
   18.                 XX'=    .987475 * SPEED * SIN(HEADING) ;
   19.                 YY'=    .987475 * SPEED * COS(HEADING);
   20.
   20.                         "FIND SIN AND COS OF THE TWO COORDINATE FRAMES"
   20.                 SIN'PHI'TRACK=ATAN(.99327733 * TAN(LAT'TRACK));
   21.                 SIN'PHI'CENTER=ATAN(.99327733 * TAN(LAT'CENTER));
   22.                 COS'PHI'TRACK=SQRT(1. - (SIN'PHI'TRACK)**2.);
   23.                 COS'PHI'CENTER=SQRT(1. - (SIN'PHI'CENTER)**2.);
   24.
   24.                         "FIND EARTH RADIUS"
   24.                 EARTH'RADIUS = 3444.054 * COS(LAT'CENTER)
   24.                              / (COS'PHI'CENTER * SQRT(1. - .00672267 * SIN(LAT'CENTER)**2.));
   25.
   25.                         "FIND TRACK POSITION"
   25.                 DIVISOR = 1. + SIN'PHI'CENTER * SIN'PHI'TRACK
   25.                         + COS'PHI'CENTER * COS'PHI'TRACK * COS(LONG'CENTER);
   25.
   26.                         "X TRACK POSITION"
   26.                 XX = (2. * EARTH'RADIUS * COS'PHI'TRACK
   26.                      * SIN(LONG'CENTER)) / DIVISOR;
   27.
   27.                         "Y TRACK POSITION"
   27.                 YY = (2. * EARTH'RADIUS * (SIN'PHI'TRACK * COS'PHI'CENTER
   27.                      - COS'PHI'TRACK * SIN'PHI'CENTER
   27.                        * COS(LONG'CENTER))) / DIVISOR;
   28.                 RETURN;
   29.                 END     "COORDINATE'TRANSFORMATION"
```

STATISTIC NAME	OCCURRENCES	PERCENTAGE
CHARACTERS	1292	
LINES	50	
SYMBOLS	233	
KEY WORDS	19	8.15
BEGIN	1	5.26
END	1	5.26
ITEM	15	78.95
PROC	1	5.26
RETURN	1	5.26
COMMENTS	10	4.29
DIRECTIVES	1	0.43
COMPOOL	1	100.0
CONSTANTS	16	6.87
FLOAT	15	93.75
CHARACTER	1	6.25
SIGNS	113	48.50
+	2	1.77
-	4	3.54
*	20	17.70
/	3	2.65
**	3	2.65
=	10	8.85
;	28	24.78
!	1	0.88
(	21	18.58
)	21	18.58
ABBREVIATIONS/DEFINE FORMAL PARAMETERS	15	6.44
F	15	100.0
NAMES	68	29.18
COMPOOL	1	1.47
PROC	15	22.6
SIMPLE-ITEM	52	76.47
DECLARATIONS	16	
SIMPLE-ITEM	15	93.75
NON-BASED	15	100.0
PROC	1	6.25
STATEMENTS	30	
SINGLE ASSIGNMENT	10	33.33
FUNCTION CALL	14	46.67
RETURN	1	3.33

```
STMT   LOCN R   CODE R   CODE R   LABEL   MNEM   OPERANDS

       00E8 H   0018 L                     HEADING
       00E9 H   0024 L                     L.24
       00EA H   002C L                     LAT'TRACK
       00EB H   0024 L                     L.24
       00EC H   002C L                     L.2C
       00ED H   0020 L                     L.20
       00EE H   000A L                     LAT'CENTER
       00EF H   0020 L                     L.20
       00F0 H   002C L                     L.2C
       00F1 H   0024 L                     L.24
       00F2 H   002C L                     L.2C
       00F3 H   0022 L                     L.22
       00F4 H   000A L                     LAT'CENTER
       00F5 H   0024 L                     L.24
       00F6 H   000A L                     LAT'CENTER
       00F7 H   0022 L                     L.22
       00F8 H   002C L                     L.2C
       00F9 H   001E L                     L.1E
       00FA H   0008 L                     LONG'CENTE
       00FB H   001E L                     L.1E
       00FC H   7E65   9400                0.98747499E0
       00FE H   7F23   B500                0.99327730E0
       0100 H   4000   0001                1.0E0
       0102 H   6E24   E0F9                0.67226700E-2
       0104 H   6BA0   6E0C                0.34440540E4
       0106 H   4000   0002                0.20000000E1
. REGION 1 *
2.
. REGION 2 *
       0000 H                      COORDIN
. REGION 3 *
18.
. REGION 4 *
       0002 H   83F0   00E8 H               LIM   15,(HEADING)
       0004 H   7220   0000 *               JS    2,SIN
       0006 H   8400   00FC H               DL    0,(0.98747499E0)
       0008 H   C700   001A L               FM    0,SPEED
       000A H   C700   0024 L               FM    0,L.24
       000C H   9400   0004 L               DST   0,XX'
19.
. REGION 5 *
       000E H   83F0   00E8 H               LIM   15,(HEADING)
       0010 H   7220   0000 *               JS    2,COS
       0012 H   8400   00FC H               DL    0,(0.98747499E0)
       0014 H   C700   001A L               FM    0,SPEED
       0016 H   C700   0024 L               FM    0,L.24
       0018 H   9400   0006 L               DST   0,YY'
20.
. REGION 6 *
       001A H   83F0   00EA H               LIM   15,(LAT'TRACK)
       001C H   7220   0000 *               JS    2,TAN
```

```
STMT    LOCN R    CODE R  CODE R    LABEL   NAME  OPERANDS
                  8400    00FE H            DL    0,(0.9932773@E0)
        0023 H    C700    0024 L            FM    0,L.24
        0022 H    9400    0022 L            DST   0,L.22
        0024 H    9400    002C L            DST   0,L.2C
        0026 H    83F0    00EC H            LIM   15,(L,2C)
        0028 H    7220    0000 *            JS    2,ATAN
        002A H    8400    0020 L            DL    0,L.20
        002C H    9400    0016 L            DST   0,SIN'PHI'TR
21.
 . REGION 7 *
        002E H    83F0    00EE H            LIM   15,(LAT'CENTER)
        0030 H    7220    0000 *            JS    2,TAN
        0032 H    8400    00FE H            DL    0,(0.9932773@E0)
        0034 H    C700    0024 L            FM    0,L.24
        0036 H    9400    0022 L            DST   0,L.22
        0038 H    9400    002C L            DST   0,L.2C
        003A H    83F0    00F0 H            LIM   15,(L,2C)
        003C H    7220    0000 *            JS    2,ATAN
        003E H    8400    0024 L            DL    0,L.24
        0040 H    9400    0014 L            DST   0,SIN'PHI'CE
22.
 . REGION 8 *
        0042 H    8400    0016 L            DL    0,SIN'PHI'TR
        0044 H    9400    002C L            DST   0,L.2C
        0046 H    C700    002C L            FM    0,L.2C
        0048 H    8420    0100 H            DL    2,(1.0E0)
        004A H    9400    002C L            DST   0,L.2C
        004C H    B720    002C L            FS    2,L.2C
        004E H    9420    0024 L            DST   2,L.24
        0050 H    9420    002C L            DST   2,L.2C
        0052 H    83F0    00F2 H            LIM   15,(L,2C)
        0054 H    7220    0000 *            JS    2,SQRT
        0056 H    8400    0022 L            DL    0,L.22
        0058 H    9400    0012 L            DST   0,COS'PHI'TR
23.
 . REGION 9 *
        005A H    8400    0014 L            DL    0,SIN'PHI'CE
        005C H    9400    002C L            DST   0,L.2C
        005E H    C700    002C L            FM    0,L.2C
        0060 H    8420    0100 H            DL    2,(1.0E0)
        0062 H    9400    002C L            DST   0,L.2C
        0064 H    B720    002C L            FS    2,L.2C
        0066 H    9420    0022 L            DST   2,L.22
        0068 H    9420    002C L            DST   2,L.2C
        006A H    83F0    00F4 H            LIM   15,(L,2C)
        006C H    7220    0000 *            JS    2,SQRT
        006E H    8400    0024 L            DL    0,L.24
        0070 H    9400    0018 L            DST   0,COS'PHI'CE
24.
 . REGION 10*
        0072 H    83F0    00F4 H            LIM   15,(LAT'CENTER)
        0074 H    7220    0000 *            JS    2,COS
```

```
STMT   LOCA R   CODE R   CODE P   LABEL   MNEM   OPERANDS
                0078 H  83F0  00F6 R          LIM   15,(DAT'CENTER)
                007B H  7220  0000 *          JS    2,SIN
                007A H  8400  0022 L          DL    0,L.22
                007C H  9400  002C L          DST   0,L.2C
                007E H  C700  002C L          FM    0,L.2C
                0082 H  C700  0102 H          FM    0,(0.6722670E-2)
                0082 H  8420  0100 H          DL    2,(1.0E0)
                0084 H  9400  002C L          DST   0,L.2C
                0086 H  8720  002C L          FS    2,L.2C
                0088 H  9420  0020 L          DST   2,L.20
                008A H  9420  002C L          DST   2,L.2C
                008C H  83F0  00F8 H          LIM   15,(L.2C)
                008E H  7220  0000 *          JS    2,SQRT
                0090 H  8400  0104 H          DL    0,(0.34440540E4)
                0092 H  C700  0024 L          FM    0,L.24
                0094 H  8420  001E L          DL    2,L.1E
                0096 H  C720  0010 L          FM    2,COS'PHI'CE
                0098 H  9420  002C L          DST   2,L.2C
                009A H  D700  002C L          FD    0,L.2C
                009C H  9400  000E L          DST   0,EARTH'RADI
25.
 . REGION 11*
                009E H  83F0  00FA H          LIM   15,(LONG'CENTE)
                00A0 H  7220  0000 *          JS    2,COS
                00A2 H  8400  0012 L          DL    0,COS'PHI'TR
                00A4 H  C700  0010 L          FM    0,COS'PHI'CE
                00A6 H  C700  001E L          FM    0,L.1E
                00A8 H  8420  0016 L          DL    2,SIN'PHI'TR
                00AA H  C720  0014 L          FM    2,SIN'PHI'CE
                00AC H  A720  0100 H          FA    2,(1.0E0)
                00AE H  9400  002C L          DST   0,L.2C
                00B0 H  A720  002C L          FA    2,L.2C
                00B2 H  9420  001C L          DST   2,DIVISOR
26.
 . REGION 12*
                00B4 H  83F0  00FA H          LIM   15,(LONG'CENTE)
                00B6 H  7220  0000 *          JS    2,SIN
                00B8 H  8400  0106 H          DL    0,(0.2000000E1)
                00BA H  C700  000E L          FM    0,EARTH'RADI
                00BC H  C700  0012 L          FM    0,COS'PHI'TR
                00BE H  C700  001E L          FM    0,L.1E
                00C0 H  D700  001C L          FD    0,DIVISOR
                00C2 H  9400  0000 L          DST   0,XX
27.
 . REGION 13*
                00C4 H  83F0  00FA H          LIM   15,(LONG'CENTE)
                00C6 H  7220  0000 *          JS    2,COS
                00C8 H  8400  0010 L          DL    0,COS'PHI'CE
                00CA H  C700  0016 L          FM    0,SIN'PHI'TR
                00CC H  8420  0014 L          DL    2,SIN'PHI'CE
                00CE H  C720  0012 L          FM    2,COS'PHI'TR
                00D0 H  C720  001E L          FM    2,L.1E
```

```
STMT    LOCN R    CODE R   CODE R   LABEL    MNEM   OPERANDS
        00D2 H    942Ø     002C L            DST    2,L.2C
        00D4 H    B7ØØ     002C L            FS     Ø,L.2C
        00D6 H    842Ø     0106 H            DL     2,(Ø.2ØØØØØØØE1)
        00D8 H    C72Ø     ØØØE L            FM     2,EARTH'RADI
        00DA H    942Ø     002C L            DST    Ø,L.2C
        00DC H    C72Ø     002C L            FM     2,L.2C
        00DE H    D72Ø     001C L            FD     2,DIVISOR
        00E0 H    942Ø     0002 L            DST    2,YY
28.
. REGION 14*
        00E2 H    70F0     00E4 H            J      H.E4
29.
. REGION 15*
        00E4 H                        H.E4
. REGION 16*
        00E4 H    8F3Ø     0026 L            LM     3,L.26
        00E6 H    70F2     0000              J      \Ø,2

        0400 H    9F3Ø     0026 L            STM    3,L.26
. REGION 17*
30.
. REGION 18*
                                      END
```

NAME	CLAS	SCOPE	LOC	TY	FB	SIZ	PERM/DSIZ	DEF	- - SET(*)/USED - -			
ATAN	PROC	*GLOBAL*	0000*	F	0	32	0	2:	20	21		
COORDINATE	PROC	*GLOBAL*	0000H				16	2:				
COS	PROC	*GLOBAL*	0000*	F	0	32	0	2:	19	24	25	27
COS'PHI'CE	ITEM	COORDINATE	0010L	F	0	32	RESERV	11:	23*	24	25	27
COS'PHI'TR	ITEM	COORDINATE	0012L	F	0	32	RESERV	12:	22*	25	26	27
DIVISOR	ITEM	COORDINATE	001CL	F	0	32	RESERV	17:	25*	26	27	
EARTH'RADI	ITEM	COORDINATE	000EL	F	0	32	RESERV	10:	24*	26	27	
HEADING	ITEM	COORDINATE	0018L	F	0	32	RESERV	15:	18	19		
JTIDS	BLOC	*GLOBAL*	0000*			0	EXTRNL	2:				
LAT'CENTER	ITEM	COORDINATE	000AL	F	0	32	RESERV	8:	21	24	24	
LAT'TRACK	ITEM	COORDINATE	000CL	F	0	32	RESERV	9:	20			
LONG'CENTE	ITEM	COORDINATE	0008L	F	0	32	RESERV	7:	25	26	27	
ON	DEFN	*GLOBAL*						2:				
OUTPUT	PROC	*GLOBAL*	0000				3	2:				
PERFORM	DEFN	*GLOBAL*						2:				
READR	PROC	*GLOBAL*	0000				0	2:				
SIN	PROC	*GLOBAL*	0000*	F	0	32	0	2:	18	24	26	
SIN'PHI'CE	ITEM	COORDINATE	0014L	F	0	32	RESERV	13:	21*	23	25	27
SIN'PHI'TR	ITEM	COORDINATE	0016L	F	0	32	RESERV	14:	20*	22	25	27
SPEED	ITEM	COORDINATE	001AL	F	0	32	RESERV	16:	18	19		
SQRT	PROC	*GLOBAL*	0000*	F	0	32	0	2:	22	23	24	
TAN	PROC	*GLOBAL*	0000	F	0	32	0	2:	20	21		
WRITEW	PROC	*GLOBAL*	0000				0	2:				
XX	ITEM	COORDINATE	0000L	F	0	32	RESERV	3:	26*			
XX'	ITEM	COORDINATE	0004L	F	0	32	RESERV	5:	18*			
YY	ITEM	COORDINATE	0002L	F	0	32	RESERV	4:	27*			
YY'	ITEM	COORDINATE	0006L	F	0	32	RESERV	6:	19*			

PROGRAM SUMMARY
DATA/VARIABLES 0400 - 002D
INSTRUCTIONS/CONSTANTS 8400 - 8107
EXTERNS: TAN SQRT SIN COS JTIDS ATAN
INTERNS: COORDI
FILES REFERENCED:
JTIDS.CMP 9/14/76 9:36 CMP:JTIDS

50 LINES 3 MESSAGES: 3 INFORMATION
CPU TIME 3.866 SEC

APPENDIX B

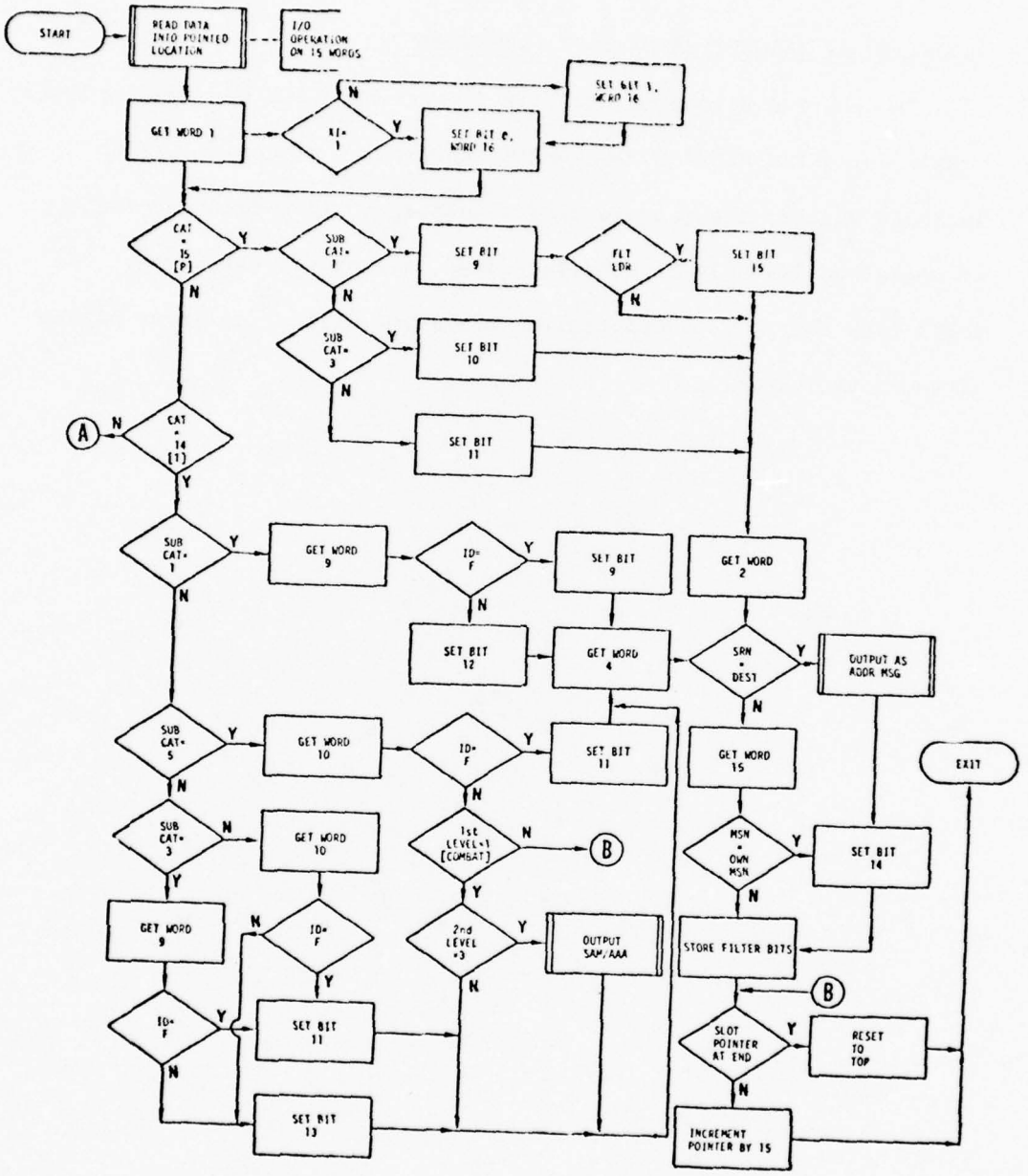
ALGORITHM #2, ACCEPT/HASH/STORE

(ORIGINAL JTIDS FLOW CHART)

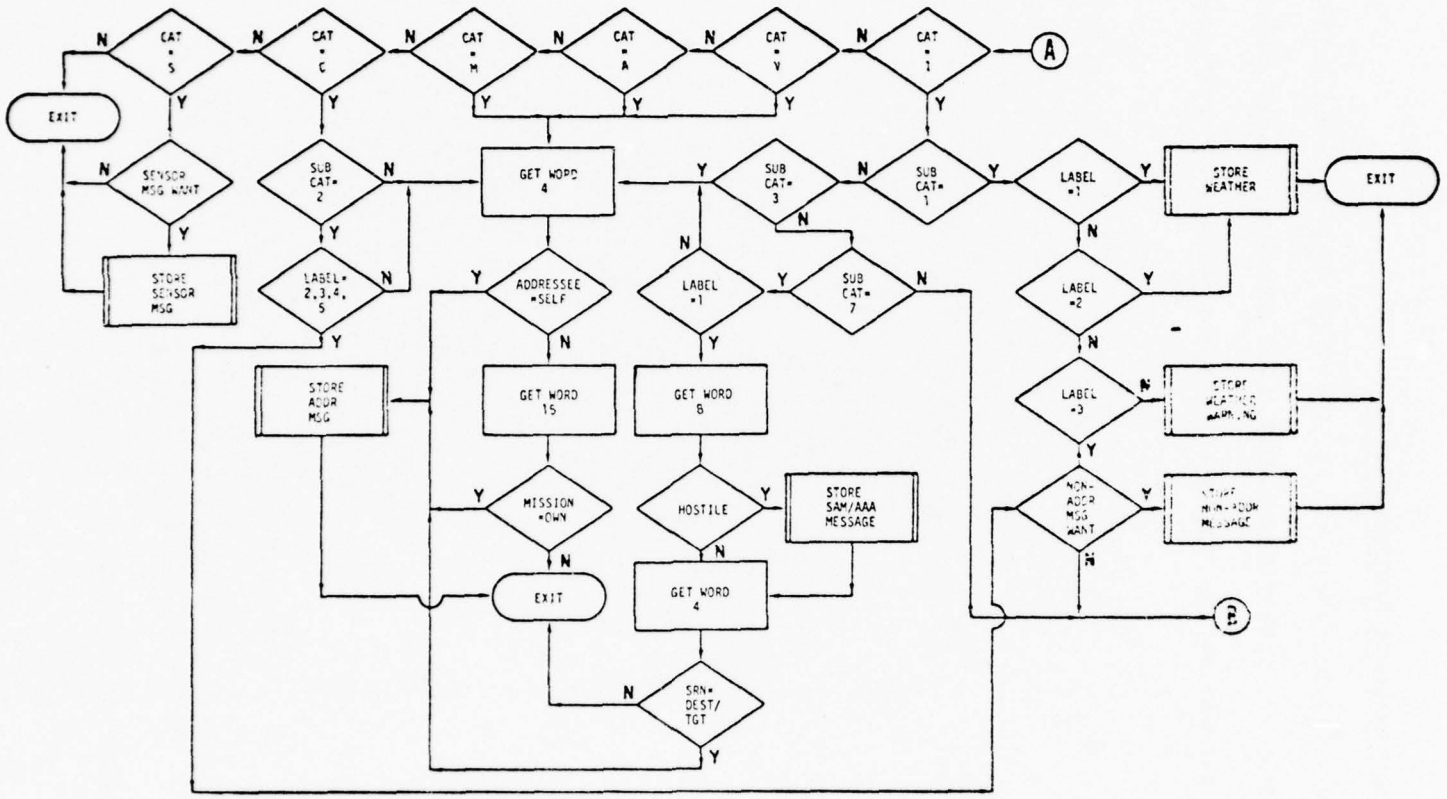
ALGORITHM #2 (ACCEPT/HASH/STORE) DESCRIPTION:

The charted algorithm examines a received JTIDS message for the LIBRARY or INPUT MESSAGE MANAGEMENT functions. It codes bits to indicate whether the message uses simulated or live data and whether it contained friendly or hostile ground, air or sea data, etc. The logic flow chart, as received from the JTIDS Office, is shown on the attached two pages.

ACCEPT, HASH, STORE

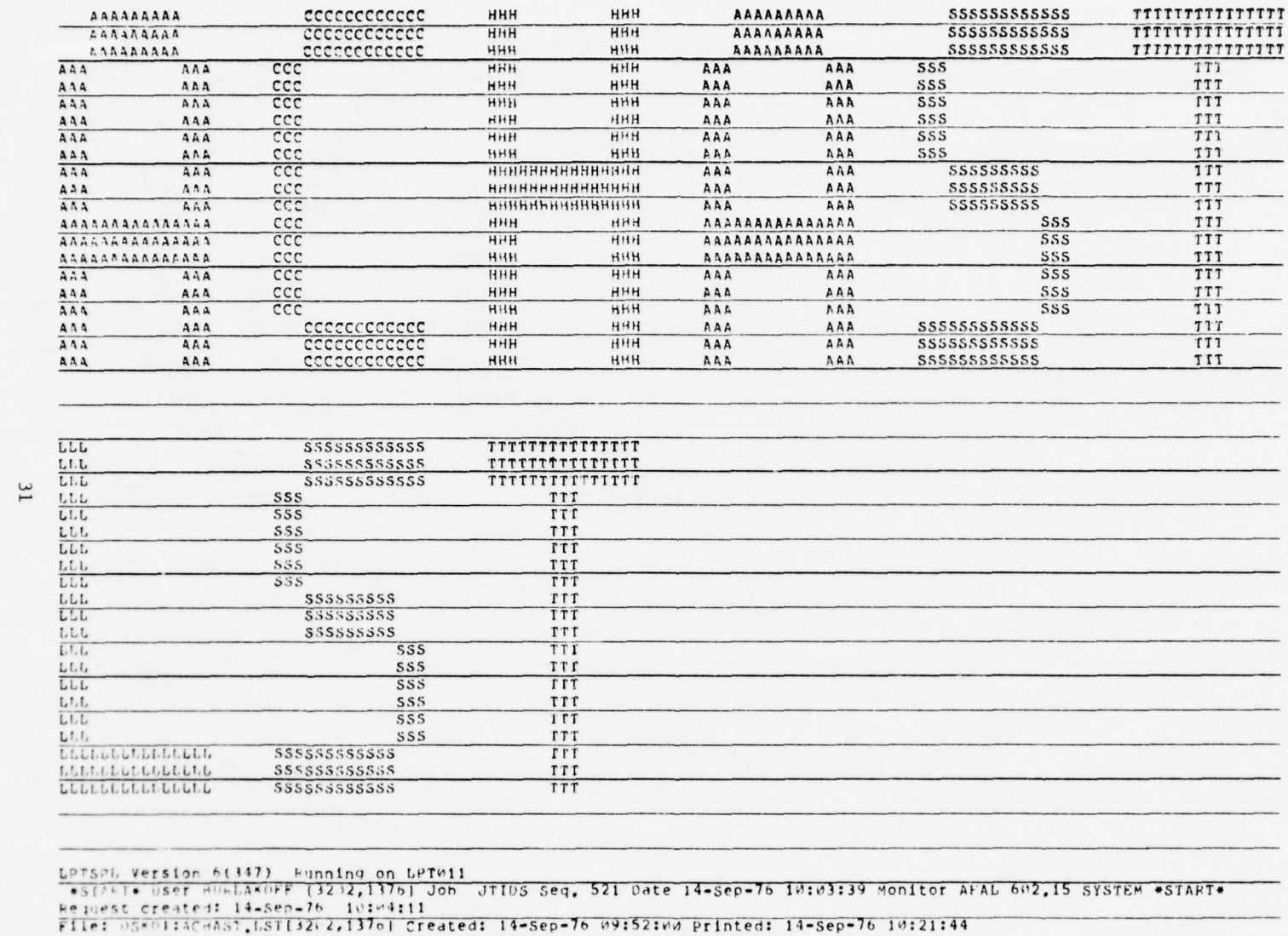


ACCEPT, HASH, STORE
(continued)



ASSEMBLY LANGUAGE IMPLEMENTATION

The following pages are the resultant assembler output for Algorithm #2, ACCEPT/HASH/STORE. The procedure (named ACHAST) is coded exactly as the original JTIDS specification (flowchart) dictated.


AAAAAAAAA CCCCCCCCCCCC HHH HHH AAAAAAAAA SSSSSSSSSSS TTTTTTTTTTTTTTT
AAA AAA CCC HHH HHH AAA AAA SSS TTT
LLL SSSSSSSSSSS TTTTTTTTTTTTTTT
SSS TTT
SSSSSSSSS
SSS
LLLLLLLLLLLLLLL SSSSSSSSSSS
LPTSPL Version 6(347) Running on LPT011
START User HUKLAKOFF [3232,137617] Job JTIDS Seq. 521 Date 14-Sep-76 10:03:39 Monitor APAL 602,15 SYSTEM *START*
Request created: 14-Sep-76 10:04:11
File: DSK01:ACHAST.LST[3232,137617] Created: 14-Sep-76 09:52:00 Printed: 14-Sep-76 10:21:44
QUEUE Switches: /PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:316 /FORMS:NORMAL
File will be deleted after printing

```
*****MEMORY ALLOCATION TABLE FOR  ACHAST   ****

REFERENCE      DECIMAL MEMORY LOCATIONS          EQUIVALENT HEX MEM LOCATIONS
 INDEX      FROM        TO        TYPE        FROM        TO        TYPE
   1          0         49         S          0000        0031       S
   2         50        429         C          0032        01AD       C
```



```
LINE   LOCATION      HEX CODE A   MANUSCRIPT

   1                             MODULE  ACHAST
   2                          .
   3                          .  ACCEPT,HASH,STORE ROUTINE
   4                          .
   5                          .  REGISTER DEFINES
   6                          .
   7                          R0   EQU   0
   8                          R1   EQU   1
   9                          R2   EQU   2
  10                          R3   EQU   3
  11                          R4   EQU   4
  12                          R5   EQU   5
  13                          R6   EQU   6
  14                          R7   EQU   7
  15                          R8   EQU   8
  16                          R9   EQU   9
  17                          R10  EQU   10
  18                          R11  EQU   11
  19                          R12  EQU   12
  20                          R13  EQU   13
  21                          R14  EQU   14
  22                          R15  EQU   15
  23                          .
  24                          .  INTERNAL DECLARATIONS
  25                          .
  26                             ENTRY   ACHAST
  27                          .
  28                          .  EXTERNAL DECLARATIONS
  29                          .
  30                             EXTERNAL        READ
  31                             EXTERNAL        WRITE
  32                             EXTERNAL        OUTPUT
  33                          .
  34                          .  CONDITION DEFINES
  35                          .
  36                          LT   EQU   1
  37                          EQ   EQU   2
  38                          LE   EQU   3
  39                          GT   EQU   4
  40                          NE   EQU   5
  41                          GE   EQU   6
  42                          .
  43                          .  LOCAL DATA ALLOCATION
  44                          .
```

```
   45        0000          0000          WORD     STORAGE 16
   46        0010          0000          WORD'16  STORAGE 1
   47        0011          0000          XI       STORAGE 1
   48        0012          0000          CAT      STORAGE 1
   49        0013          0000          SUB'CAT  STORAGE 1
   50        0014          0000          SRN      STORAGE 1
   51        0015          0000          LABEL    STORAGE 1
   52        0016          0000          SAM'AAA  STORAGE 1
   53        0017          0000          FILTER'B         STORAGE 1
   54        0018          0000          FLT      STORAGE 1
1    PROGRAM:  ACHAST     PAGE    2     DAIS HHC ASSEMBLER; VERSION 001

 LINE   LOCATION        HEX CODE A   MANUSCRIPT

   55        0019          0000          LDR      STORAGE 1
   56        001A          0000          MSN      STORAGE 1
   57        001B          0000          OWN'MSN  STORAGE 1
   58        001C          0000          DEST     STORAGE 1
   59        001D          0000          NON'ADDR         STORAGE 1
   60        001E          0000          ADDR'MSG         STORAGE 1
   61        001F          0000          ID       STORAGE 1
   62        0020          0000          FIRST'LE         STORAGE 1
   63        0021          0000          SECOND'L         STORAGE 1
   64        0022          0000          WEATHER  STORAGE 1
   65        0023          0000          WEATHER'         STORAGE 1
   66        0024          0000          DEST'TGT         STORAGE 1
   67        0025          0000          HOSTILE  STORAGE 1
   68        0026          0000          SELF     STORAGE 1
   69        0027          0000          OWN      STORAGE 1
   70        0028          0000          ADDRESSE         STORAGE 1
   71        0029          0000          MISSION  STORAGE 1
   72        002A          0000          SENSOR'M         STORAGE 1
   73        002B          0000          WANT     STORAGE 1
   74        002C          0000          SLOT'POI         STORAGE 1
   75        002D          0000          END'POIN         STORAGE 1
   76        002E          0000          TOP'POIN         STORAGE 1
   77        002F          0000          SAVE     STORAGE 3
   78                                    .
   79                                    .
   80                                    .        ENTRY TO MAIN PROGRAM
   81                                    .
   82                                    .
   83                                    .        SAVE REGISTERS
   84        0032  9F20    002F    R     ACHAST   STM      R2,SAVE
   85                                    .        READ IN DATA
   86        0034  83F0    0000    R              LIM      R15,WORD
   87        0036  722F    FFFF    X              JS       R2,READ
   88                                    .        LOAD XI, CHECK IF EQ 1
   89        0038  5000    0011    R              L        R0,XI
   90        003A  F3F0    0001    A              CIM      R0,1
   91        003C  702F    0040    R              JC       EQ,AA
   92                                    .        IF NE 1, SET BIT 1 IN WORD'16
   93        003E  5010    0010    R              SB       1,WORD'16
   94                                    .        SET BIT 0 IN WORD'16
   95        0040  5000    0010    R     AA       SB       0,WORD'16
   96                                    .        GET WORD[I] , STORE IN CAT
   97        0042  8010    0000    R              L        R1,WORD
   98        0044  9010    0012    R              ST       R1,CAT
   99                                    .        SEE IF CAT = 'P'
```

```
100         0046  F310  5020  A             CIM     R1,'P '
101         0048  7050  006C  R             JC      NE,NEXT'T
102                                 .       CATEGORY = ' P'
103                                 .       LOAD SUB'CAT, CHECK IF EQ 1
104         004A  8010  0013  R             L       R1,SUB'CAT
105         004C  F312  0001  A             CIM     R1,1
106         004E  7050  005C  R             JC      NE,P'SCNE1
107                                 .       SUB'CAT EQ 1, SET BIT 9
108         0050  5090  0010  R             SB      9,WORD'16
1   PROGRAM; ACHAST     PAGE    3     DAIS HBC ASSEMBLER; VERSION 001

LINE   LOCATION       HEX CODE A   MANUSCRIPT   ,

109                                 .       CHECK IF FLT EQ LDR
110         0052  8000  0018  R             L       R0,FLT
111         0054  F000  0019  R             C       R0,LDR
112         0056  7050  0066  R             JC      NE,P'WRD2
113                                 .       FLT EQ LDR, SET BIT 15 OF WORD'16
114         0058  50F0  0010  R             SB      15,WORD'16
115         005A  70F0  0066  R             J       P'WRD2
116                                 .       CHECK IF SUB'CAT EQ 3
117         005C  F310  0003  A   P'SCNE1   CIM     R1,3
118         005E  7050  0064  R             JC      NE,P'SCNE3
119                                 .       SUB'CAT EQ 3, SET BIT 10 OF WORD'16
120         0060  50A0  0010  R             SB      10,WORD'16
121         0062  70F0  0066  R             J       P'WRD2
122                                 .       SUB'CAT NE 3, SET BIT 11 OF WORD'16
123         0064  50B0  0010  R   P'SCNE3   SB      11,WORD'16
124                                 .
125                                 .       GET WORD[2] OF DATA
126         0066  8010  0001  R   P'WRD2   L       R1,WORD+1
127         0068  9010  0014  R             ST      R1,SRN
128                                 .       GOTO COMMON PROCESSING BETWEEN CAT = 'P' AND 'T'
129         006A  70F0  00C6  R             J       CAT'PT
130                                 .
131                                 .       CHECK ON CAT= 'T'
132                                 .
133         006C  F310  5420  A   NEXT'T    CIM     R1,'T '
134         006E  7050  00F4  R             JC      NE,NEXT'1
135                                 .
136                                 .       CAT = 'T'
137                                 .       CHECK IF SUB'CAT EQ 1
138         0070  8010  0013  R             L       R1,SUB'CAT
139         0072  F310  0001  A             CIM     R1,1
140         0074  7050  008A  R             JC      NE,T'SC5
141                                 .       GET WORD[9] OF DA
142         0076  8000  0009  R             L       R0,WORD+9
143         0078  9000  001F  R             ST      R0,ID
144                                 .       CHECK IF ID EQ 'F'
145         007A  F300  4620  A             CIM     R0,'F '
146         007C  7050  0082  R             JC      NE,T'IDNEF
147                                 .       SET BIT 9 OF WORD'16
148         007E  5090  0010  R             SB      9,WORD'16
149         0080  70F0  0084  R             J       T'WRD4
150                                 .       SET BIT 12 OF WORD'16
151         0082  50C0  0010  R   T'IDNEF   SB      12,WORD'16
152                                 .       GET WORD[4] OF DATA
153         0084  8000  0004  R   T'WRD4    L       R0,WORD+4
154         0086  9000  0014  R             ST      R0,SRN
```

```
155     0088  70F0  00C6  R            J       CAT'PT
156                                   .       CHECK IF SUB'CAT EQ 5
157     008A  F310  0005  A    T'SC5   CIM     R1,5
158     008C  7050  00AC  R            JC      NE,T'SC3
159                                   .       GET WORD[10] OF DATA
160     008E  8000  000A  R            L       R0,WORD+10
161     0090  9000  001F  R            ST      R0,ID
162                                   .       CHECK IF ID EQ 'F'
163     0092  F300  4620  A            CIM     R0,'F '
1    PROGRAM:  ACHAST      PAGE  4     DAIS HBC ASSEMBLER; VERSION 001

LINE   LOCATION      HEX CODE A   MANUSCRIPT  ,

164     0094  7050  009A  R            JC      NE,T'FL
165                                   .       IF EQ 'F', SET BIT 11 WORD'16
166     0096  5080  0010  R            SB      11,WORD'16
167     0098  70F0  0084  R            J       T'WRD4
168                                   .       CHECK IF FIRST'LEVEL NE 1
169     009A  8000  0020  R    T'FL    L       R0,FIRST'LE
170     009C  F300  0001  A            CIM     R0,1
171     009E  7050  00DE  R            JC      NE,COM'B
172                                   .       CHECK IF SECOND'LEVEL EQ 3
173     00A0  8000  0021  R            L       R0,SECOND'L
174     00A2  F300  0003  A            CIM     R0,3
175     00A4  7050  0084  R            JC      NE,T'WRD4
176                                   .       STORE SAM/AAA
177     00A6  83F0  0016  R            LIM     R15,SAM'AAA
178     00A8  7220  FFFF  X            JS      2,WRITE
179     00AA  70F0  0084  R            J       T'WRD4
180                                   .       CHECK IF SUB'CAT EQ 3
181     00AC  F310  0003  A    T'SC3   CIM     R1,3
182     00AE  7050  00B6  R            JC      NE,T'NESC3
183                                   .       GET WORD[9] OF DATA
184     00B0  8000  0009  R            L       R0,WORD+9
185     00B2  9000  001F  R            ST      R0,ID
186     00B4  70F0  00BA  R            J       T'IDF
187                                   .       GET WORD[10] OF DATA
188     00B6  8000  000A  R    T'NESC3 L       R0,WORD+10
189     00B8  9000  001F  R            ST      R0,ID
190                                   .       CHECK IF ID EQ 'F'
191     00BA  F300  4620  A    T'IDF   CIM     R0,'F '
192     00BC  7050  00C2  R            JC      NE,T'NEIDF
193                                   .       SET BIT 11 OF WORD'16
194     00BE  5080  0010  R            SB      11,WORD'16
195     00C0  70F0  0084  R            J       T'WRD4
196                                   .       SET BIT 13 OF WORD'16
197     00C2  5020  0010  R    T'NEIDF SB      13,WORD'16
198     00C4  70F0  0084  R            J       T'WRD4
199                                   .
200                                   .       CAT'PT
201                                   .       R0=SRN
202     00C6  F000  001C  R    CAT'PT  C       R0,DEST
203     00C8  7050  00D0  R            JC      NE,PT'NED
204                                   .       IF SRN EQ DEST, OUTPUT ADDR'MSG,SET BIT 14 OF WORD'16
205     00CA  5040  0010  R            SB      14,WORD'16
206     00CC  83F0  001E  R            LIM     15,ADDR'MSG
207     00CE  7220  FFFF  X            JS      2,OUTPUT
208                                   .       GET WORD[15] OF DATA
209     00D0  8000  000F  R    PT'NED  L       R0,WORD+15
```

```
 210                                        .       CHECK IF OWN'MSG
 211        00D2  9000  001A  R                     ST      R0,MSN
 212        00D4  F000  0018  R                     C       R0,OWN'MSN
 213        00D6  7050  00DA  R                     JC      NE,PT'FB
 214                                        .       IF OWN'MSG, SET BIT 14 OF WORD'16
 215        00D8  58E0  0010  R                     SB      14,WORD'16
 216                                        .       STORE   FILTER'BITS
 217        00DA  83F0  0017  R        PT'FB        LIM     R15,FILTER'B
1    PROGRAM:  ACHAST      PAGE    5     DAIS HBC ASSEMBLER; VERSION 001

 LINE   LOCATION        HEX CODE A    MANUSCRIPT    ,


 218        00DC  7220  00A9  X                     JS      2,WRITE
 219                                        .
 220                                        .       COM'B
 221                                        .
 222                                        .       CHECKS IF SLOT'POINTER AT END
 223        00DE  8000  002C  R        COM'B        L       R0,SLOT'POI
 224        00E0  F000  002D  R                     C       R0,END'POIN
 225        00E2  7050  00EA  R                     JC      NE,B'PT
 226                                        .       RESET SLOT'POINTER TO TOP
 227        00E4  8000  002E  R                     L       R0,TOP'POIN
 228        00E6  9000  002C  R                     ST      R0,SLOT'POI
 229        00E8  7000  00F0  R                     J       OUT
 230                                        .       ADD 15 TO SLOT'POINTER
 231        00EA  8000  000F  A        B'PT         LIM     R0,15
 232        00EC  A000  002C  R                     A       R0,SLOT'POI
 233        00EE  9000  002C  R                     ST      R0,SLOT'POI
 234                                        .
 235                                        .       EXIT FROM ACHAST
 236                                        .
 237        00F0  8F20  002F  R        OUT          LM      R2,SAVE
 238        00F2  7CF2  0000  A                     J       0,R2
 239        00F4  F310  4920  A        NEXT'I       CIM     R1,'I '
 240        00F6  7050  014E  R                     JC      NE,NEXT'VA
 241                                        .
 242                                        .       CATEGORY I
 243                                        .
 244        00F8  8020  0013  R                     L       R0,SUB'CAT
 245        00FA  F300  0001  A                     CIM     R0,1
 246        00FC  7050  0124  R                     JC      NE,I'SC3
 247                                        .       SUBCAT = 1
 248                                        .       CHECK IF LABEL = 1 OR 2
 249        00FE  8000  0015  R                     L       R0,LABEL
 250        0100  F300  0001  A                     CIM     R0,1
 251        0102  7020  0108  R                     JC      EQ,I'LB12
 252        0104  F300  0002  A                     CIM     R0,2
 253        0106  7050  010E  R                     JC      NE,I'LB3
 254                                        .       LABEL = 1 OR 2
 255                                        .       STORE WEATHER
 256        0108  83F0  0022  R        I'LB12       LIM     R15,WEATHER
 257        010A  7220  00DD  X                     JS      2,WRITE
 258                                        .       EXIT
 259        010C  7000  00F0  R                     J       OUT
 260                                        .       CHECK IF LABEL = 3
 261        010E  F300  0003  A        I'LB3        CIM     R0,3
 262        0110  7050  0116  R                     JC      NE,I'LNE3
 263                                        .       LABEL = 3
 264                                        .       CHECK ADDR'MSG
```

```
265        0112   8000  001D  R              L       R0,NON'ADDR
266        0114   F000  002B  R              C       R0,WANT
267                                    .      IF NON'ADDR NE WANT GOTO COM'B
268        0116   705A  00DE  R              JC      NE,COM'B
269                                    .      WANT NON'ADDR'MSG
270                                    .      STORE NON'ADDR
271        0118   83E0  001D  R              LIM     R15,NON'ADDR
1    PROGRAM;  ACHAST      PAGE   6     DAIS HBC ASSEMBLER; VERSION 001

LINE   LOCATION      HEX CODE A   MANUSCRIPT

272        011A   7220  010B  X              JS      R2,WRITE
273                                    .      EXIT
274        011C   70E0  00E0  R              J       OUT
275                                    .      LABEL NE 1, 2 OR 3, STORE WEATHER WARNING
276        011E   83E0  0023  R     I'LNE3   LIM     R15,WEATHER'
277        0120   7220  011B  X              JS      R2,WRITE
278                                    .      EXIT
279        0122   70E0  00E0  R              J       OUT
280                                    .      CHECK IF SUB'CAT = 3
281        0124   F300  0003  A     I'SC3    CIM     R0,3
282        0126   705A  012A  R              JC      NE,I'SC7
283                                    .      SUBCAT=3, GOTO CAT'VAM
284        0128   70E0  015A  R              J       CAT'VAM
285                                    .      CHECK IF SUB'CAT=7
286        012A   F300  0007  A     I'SC7    CIM     R0,7
287        012C   705A  00DE  R              JC      NE,COM'B
288                                    .      SUBCAT=7, CHECK IF LABEL=1
289        012E   8000  0015  R              L       R0,LABEL
290        0130   F300  0001  A              CIM     R0,1
291        0132   705A  015A  R              JC      NE,CAT'VAM
292                                    .      LABEL=1, GET WORD[8] OF DATA
293        0134   8010  0008  R              L       R1,WORD+8
294        0136   9010  0025  R              ST      R1,HOSTILE
295                                    .      CHECK IF HOSTILE
296        0138   F310  0001  A              CIM     R1,1
297        013A   705A  0144  R              JC      NE,I'WRD4
298                                    .      HOSTILE=ON, STORE SAM/AAA
299        013C   83E0  0016  R              LIM     R15,SAM'AAA
300        013E   7220  0121  X              JS      R2,WRITE
301                                    .      GET WORD[4] OF DATA
302        0140   8010  0003  R     I'WRD4   L       R1,WORD+3
303        0142   9010  0014  R              ST      R1,SRN
304                                    .      SEE IF SRN=DEST
305        0144   F010  0024  R              C       R1,DEST'TGT
306        0146   705A  00E0  R              JC      NE,OUT
307                                    .      SRN=DEST, STORE ADDR'MSG
308        0148   83E0  001E  R              LIM     R15,ADDR'MSG
309        014A   7220  013F  X              JS      R2,WRITE
310                                    .      EXIT
311        014C   70E0  00E0  R              J       OUT
312                                    .      SEE IF CATEGORY IS V A OR M
313        014E   F310  562A  A     NEXT'VA  CIM     R1,'V  '
314        0150   7020  015A  R              JC      EQ,CAT'VAM
315        0152   F310  4120  A              CIM     R1,'A  '
316        0154   7020  015A  R              JC      EQ,CAT'VAM
317        0156   F310  4D20  A              CIM     R1,'M  '
318        0158   705A  0176  R              JC      NE,NEXT'C
319                                           CAT =V A OR M OR COME FROM I OR C
```

```
320                                     .       GET WORD[4] OF DATA
321     015A  6000  0003  R   CAT'VAM   L       R0,WORD+3
322     015C  9600  0028  R             ST      R0,ADDRESSE
323                                     .       CHECK IF ADDRESSE IS SELF
324     015E  F000  0026  R             C       R0,SELF
325     0160  7050  0168  R             JC      NE,VAM'NES
1   PROGRAM:  ACHAST      PAGE   7     DAIS HBC ASSEMBLER; VERSION 001

 LINE   LOCATION       HEX CODE A    MANUSCRIPT    .


326                                     .       ADDRESSEE IS SELF, STORE ADDR'MSG
327     0162  83F0  001E  R             LIM     R15,ADDR'MSG
328     0164  7220  014B  X             JS      R2,WRITE
329                                     .       EXIT
330     0166  70F0  00F0  R             J       OUT
331                                     .       ADDRESSEE NOT SELF, GET WORD[15] OF DATA
332     0168  8000  000E  R   VAM'NES   L       R0,WORD+14
333                                     .       CHECK IF MISSION OWN
334     016A  8010  0029  R             L       R1,MISSION
335     016C  F010  0027  R             C       R1,OWN
336     016E  7050  00F0  R             JC      NE,OUT
337                                     .       MISSSION IS OWN,STORE ADDR'MSG
338     0170  83F0  001E  R             LIM     R15,ADDR'MSG
339     0172  7220  0165  X             JS      R2,WRITE
340                                     .       EXIT
341     0174  70F0  00F0  R             J       OUT
342     0176  F300  4320  A   NEXT'C    CIM     R0,'C '
343     0178  7050  019E  R             JC      NE,NEXT'S
344                                     .       CATEGORY IS C
345                                     .       CHECK IF SUB'CAT=2
346     017A  8000  0013  R             L       R0,SUB'CAT
347     017C  F300  0002  A             CIM     R0,2
348     017E  7050  015A  R             JC      NE,CAT'VAM
349                                     .       SUB'CAT=2, CHECK IF LABEL =2,3,4OR 5
350     0180  8000  0015  R             L       R0,LABEL
351     0182  F300  0002  A             CIM     R0,2
352     0184  7020  0192  R             JC      EQ,C'LAB
353     0186  F300  0003  A             CIM     R0,3
354     0188  7020  0192  R             JC      EQ,C'LAB
355     018A  F300  0004  A             CIM     R0,4
356     018C  7020  0192  R             JC      EQ,C'LAB
357     018E  F300  0005  A             CIM     R0,5
358     0190  7050  015A  R             JC      NE,CAT'VAM
359                                     .       LABEL=2,3,4 OR 5
360                                     .       CHECK IF NON'ADDR WANTED
361     0192  8010  001D  R   C'LAB     L       R1,NON'ADDR
362     0194  F010  002B  R             C       R1,WANT
363     0196  7050  00DE  R             JC      NE,COM'B
364                                     .       STORE NAN'ADDR
365     0198  83F0  001D  R             LIM     R15,NON'ADDR
366     019A  7220  0173  X             JS      R2,WRITE
367     019C  70F0  00F0  R             J       OUT
368                                     .       CHECK IF CATEGORY IS 'S'
369     019E  F300  5320  A   NEXT'S    CIM     R0,'S '
370     01A0  7050  00F0  R             JC      NE,OUT
371                                     .       CATEGORY IS S
372                                     .       CHECK TO STORE SENSOR'MSG
373     01A2  8000  002A  R             L       R0,SENSOR'M
374     01A4  F000  002B  R             C       R0,WANT
```

```
375       01A6   7050  00F0   R            JC      NE,OUT
376                                   .    STORE SENSOR'MSG
377       01A8   83F0  002A   R            LIM     R15,SENSOR'M
378       01AA   7220  019A   X            JS      R2,WRITE
379                                   .    EXIT
380       01AC   70F0  00F0   R            J       OUT
1   PROGRAM;  ACHAST    PAGE    8     DAIS HBC ASSEMBLER; VERSION 001

LINE   LOCATION      HEX CODE A   MANUSCRIPT

381                                        END

 .  NO START ADDRESS ENCOUNTERED

 0  ERROR(S) THIS RUN.
```

1 SYMBOLS	** ASSIGNMENT ** HEX	DECIMAL	LINE	MUL DEF	ATTRIBUTE
AA	0040	64	95		R
ACHAST	0032	50	84		RE
ADDRESSE	0028	40	70		R
ADDR'MSG	001E	30	60		R
B'PT	00EA	234	231		R
CAT	0012	18	48		R
CAT'PT	00C6	198	202		R
CAT'VAM	015A	346	321		R
COM'B	00DE	222	223		R
C'GAB	0192	402	361		R
DEST	001C	28	58		R
DEST'TGT	0024	36	66		R
END'POIN	002D	45	75		R
EQ	0002	2	37		A
FILTER'B	0017	23	53		R
FIRST'LE	0020	32	62		R
FLT	0018	24	54		R
GE	0006	6	41		A
GT	0004	4	39		A
HOSTILE	0025	37	67		R

ID	001F	31	61		R
I'LB12	0108	264	256		R
I'LB3	010E	270	261		R
I'LBE3	011E	286	276		R
I'SC3	0124	292	281		R
I'SC7	012A	298	286		R
I'WRD2	0140	320	302		R

I SYMBOLS	** ASSIGNMENT ** HEX	DECIMAL	LINE	MUL DEF	ATTRIBUTE
LABEL	0015	21	51		R
LDR	0019	25	55		R
LE	0003	3	38		A
LT	0001	1	36		A
MISSION	0029	41	71		R
MSN	001A	26	56		R
NE	0005	5	40		A
NEXT'VA	014E	334	313		R
NEXT'I	006C	108	133		R
NEXT'C	0176	374	342		R
NEXT'I	02F4	244	239		R
NEXT'S	019F	414	369		R
NON'ADDR	001D	29	59		R
OUT	02F0	240	237		R
OUTPUT	03CF	207	32		EX
OWN	0027	39	69		R
OWN'MSN	001B	27	57		R
PI'FB	00DA	218	217		R
PI'NED	00D0	208	209		R
P'SCNE3	0064	100	123		R
P'SCNE1	005C	92	117		R
P'WRD2	0066	102	126		R

READ	0037	55	30		EX
R0	0000	0	7		A
R1	0001	1	8		A
R10	000A	10	17		A
R11	000B	11	18		A

1 SYMBOLS	** ASSIGNMENT ** HEX	DECIMAL	LINE	MUL DEF	ATTRIBUTE
R12	000C	12	19		A
R13	000D	13	20		A
R14	000E	14	21		A
R15	000F	15	22		A
R2	0002	2	9		A
R3	0003	3	10		A
R4	0004	4	11		A
R5	0005	5	12		A
R6	0006	6	13		A
R7	0007	7	14		A
R8	0008	8	15		A
R9	0009	9	16		A
SAM'AAA	0016	22	52		R
SAVE	002F	47	77		R
SECOND'L	0021	33	63		R
SELF	0026	38	68		R
SENSOR'A	002A	42	72		R
SLOT'POI	002C	44	74		R
SR0	0014	20	50		R
SUB'CAT	0013	19	49		R
TOP'POIN	002E	46	76		R
T'EL	009A	154	169		R
T'IDF	00BA	186	191		R
T'IDFF	00B2	178	151		R

T'NEIDF	00C2	194	197		R
T'NESC3	00B6	182	188		R
T'SC3	00AC	172	181		R

I SYMBOLS	** ASSIGNMENT ** HEX	DECIMAL	LINE	MUL DEF	ATTRIBUTE
T'SC5	008A	138	157		R
T'WRD4	0064	132	153		R
VAM'NES	0168	360	332		R
WAIT	002B	43	73		R
WEATHER'	0023	35	65		R
WEATHER	0022	34	64		R
WORD	0000	0	45		R
WORD'16	0010	16	46		R
WRITE	01A6	427	31		EX
XI	0011	17	47		R

ELAPSED TIME = 185 SECONDS

JOVIAL-73/I IMPLEMENTATION

The following pages are the resultant compiler output for the JOVIAL-73/I compilation of Algorithm #2, ACCEPT/HASH/STORE. This procedure (named MBLT) is coded exactly as the original JTIDS specification (flow chart) dictated.

```
MMM         MMM   BBBBBBBBBBBBBB          LLL                 TTTTTTTTTTTTTTT
MMM         MMM   BBBBBBBBBBBBBB          LLL                 TTTTTTTTTTTTTTT
MMM         MMM   BBB         BBB         LLL                 TTTTTTTTTTTTTTT
MMMMMM   MMMMMM   BBB         BBB         LLL                       TTT
MMMMMMM  MMMMMMM  BBB         BBB         LLL                       TTT
MMM MMMMMMM MMM   BBB         BBB         LLL                       TTT
MMM  MMMMM  MMM   BBB         BBB         LLL                       TTT
MMM   MMM   MMM   BBB         BBB         LLL                       TTT
MMM   MMM   MMM   BBB         BBB         LLL                       TTT
MMM         MMM   BBBBBBBBBBBBBB          LLL                       TTT
MMM         MMM   BBBBBBBBBBBBBB          LLL                       TTT
MMM         MMM   BBBBBBBBBBBBBB          LLL                       TTT
MMM         MMM   BBB         BBB         LLL                       TTT
MMM         MMM   BBB         BBB         LLL                       TTT
MMM         MMM   BBB         BBB         LLL                       TTT
MMM         MMM   BBB         BBB         LLL                       TTT
MMM         MMM   BBB         BBB         LLL                       TTT
MMM         MMM   BBBBBBBBBBBBBB          LLLLLLLLLLLLLLLL          TTT
MMM         MMM   BBBBBBBBBBBBBB          LLLLLLLLLLLLLLLL          TTT
MMM         MMM   BBBBBBBBBBBBBB          LLLLLLLLLLLLLLLL          TTT


LLL                    SSSSSSSSSSS      TTTTTTTTTTTTTTT
LLL                    SSSSSSSSSSS      TTTTTTTTTTTTTTT
LLL                    SSSSSSSSSSS      TTTTTTTTTTTTTTT
LLL              SSS                        TTT
LLL              SSS                        TTT
LLL              SSS                        TTT
LLL              SSS                        TTT
LLL              SSS                        TTT
LLL              SSS                        TTT
LLL                 SSSSSSSSS               TTT
LLL                 SSSSSSSSS               TTT
LLL                 SSSSSSSSS               TTT
LLL                        SSS              TTT
LLL                        SSS              TTT
LLL                        SSS              TTT
LLL                        SSS              TTT
LLL                        SSS              TTT
LLL                        SSS              TTT
LLLLLLLLLLLLLLL  SSSSSSSSSSS                TTT
LLLLLLLLLLLLLLL  SSSSSSSSSSS                TTT
LLLLLLLLLLLLLLL  SSSSSSSSSSS                TTT

LPTSPL Version 6(317)  Running on LPTV11
 *START* User BURLAKOFF [32,2,1376] Job  MBLT Seq. 566 Date 14-Sep-76 14:40:46 Monitor AFAL 6.2.15 SYSTEM *START*
Request created: 14-Sep-76  14:41:28
File: DSKB1:MBLT.LST[32,2,1376] Created: 14-Sep-76 14:39:00 Printed: 14-Sep-76 14:41:02
QUEUE Switches:  /PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:87 /FORMS:NORMAL
File will be deleted after printing
```

```
MBLT,MBLT=MBLT.ASM/HBC/MAC/ACR/NOIN/STAT

    1.          !COMPOOL('ITIDS.CMP');
    2.          PROC    ACCEPT'HASH'STORE;
    3.                  BEGIN   "ACCEPT'HASH'STORE"
    3.
    3.                  "DECLARATION SECTION"
    3.
    3.                  "EXTERNAL DECLARATIONS"
    3.
    3.                  ITEM    WORD'16 U;
    4.
    4.                  "LOCAL DECLARATIONS"
    4.
    4.                  TABLE   DATA[1:15]      1;
    5.                          ITEM    WORD    C 2[0,0];
    6.                  ITEM    XI      U;
    7.                  ITEM    CAT     C 2;
    8.                  ITEM    SUB'CAT U;
    9.                  ITEM    FLT     U;
   10.                  ITEM    SRN     U;
   11.                  ITEM    ID      U;
   12.                  ITEM    FIRST'LEVEL     U;
   13.                  ITEM    SECOND'LEVEL    U;
   14.                  ITEM    NON'ADDR'MSG    U;
   15.                  ITEM    ADDRESSE        U;
   16.                  ITEM    LABEL   U;
   17.                  ITEM    HOSTILE U;
   18.                  ITEM    MSN     U;
   19.                  ITEM    MISSION U;
   20.
   20.                  ITEM    LDR     U;
   21.                  ITEM    WANT    U;
   22.                  ITEM    DEST    U;
   23.                  ITEM    DEST'TGT        U;
   24.                  ITEM    SENSOR'MSG      U;
   25.                  ITEM    OWN'MSN U;
   26.                  ITEM    SELF    U;
   27.                  ITEM    OWN     U;
   28.
   28.                  ITEM    SAM'AAA U;
   29.                  ITEM    FILTER'BITS     U;
   30.                  ITEM    ADDR'MSG        U;
   31.                  ITEM    WEATHER U;
   32.                  ITEM    WEATHER'WARNING U;
   33.
   33.                  ITEM    SLOT'POINTER    U;
   34.                  ITEM    END'POINTER     U;
   35.                  ITEM    TOP'POINTER     U;
   36.                  GOTO MAIN;      "GOES TO MAIN PROGRAM"
   37.
   37.          !EJECT;
```

```
38.                "                                                              "
38.                "THIS IS COMMON BETWEEN CATEGORY P AND T                       "
38.                "IT CHECKS IF SRN EQ DEST, IF SO OUTPUTS ADDR'MSG AND SETS BIT 14      "
38.                "  IF NOT, GETS WORD 15, CHECKS IF OWN'MSN,  IF SO, SETS BIT 14"
38.                "IT STORES FILTER'BITS AND PERFORMS BLK'10                     "
38.                "                                                              "
38.
38.          BLK'8:          IF  SRN = DEST;
39.                              "THEN"
39.                                  BEGIN   "SRN EQ DEST"
39.                                  OUTPUT(ADDR'MSG);       "OUTPUT ADDR'MSG"
40.                                  BIT(WORD'16,14)=1;      "SET BIT 14"
41.                                  END     "SRN EQ DEST"
42.                              ELSE
42.                                  BEGIN   "SRN NE DEST"
43.                                  MSN=WORD[15];   "GET WORD 15"
44.                                  IF  MSN = OWN'MSN;
45.                                      "THEN"
45.                                          BIT(WORD'16,14)=1;      "SET BIT 14"
46.                                  END     "SRN NE DEST"
47.                          WRITEW(FILTER'BITS);    "STORE FILTER BITS"
48.
48.          !EJECT;
```

```
49.                  "                                                    "
49.                  "THIS RESETS SLOT POINTER          "
49.                  "TO EITHER 15 MORE OR TO TOP                         "
49.                  "                                                    "
49.
49.
49.          BLK'10:                 IF  SLOT'POINTER = END'POINTER;
50.                                "THEN"
50.                                    SLOT'POINTER=TOP'POINTER;       "RESET TO TOP"
51.                                ELSE
52.                                    SLOT'POINTER=SLOT'POINTER+15;   "ADD 15"
53.
53.                            GOTO OUT;
54.                  "THIS IS COMMON BETWEEN CATEGORIES V, A, M, I, AND C     "
54.                  "IT GETS WORD 4, THEN CHECKS ADDRESSE, IF SELF, STORES ADDR'MSG,          "
54.                  "  IF NOT SELF, GETS WORD 15 THEN CHECKS MISSION, IF OWN IT STORES ADDR'MSG"
54.                  "                                                                        "
54.
54.          BLK'13:                 ADDRESSE=WORD[4];       "GET WORD 4"
55.                               IF  ADDRESSE = SELF;
56.                                "THEN"
56.                                    WRITEW(ADDR'MSG);       "STORE ADDR'MSG"
57.                                ELSE
57.                                    BEGIN   "ADDRESSE NE SELF"
58.                                    MISSION=WORD[15];       "GET WORD 15"
59.                                    IF  MISSION = OWN;
60.                                        "THEN"
60.                                        WRITEW(ADDR'MSG);       "STORE ADDR'MSG"
61.                                    END     "ADDRESSE NE SELF"
62.                               GOTO OUT;
63.
63.          !EJECT;
```

```
64.
64.                "                                                    "
64.                "    ENTRY TO MAIN PROGRAM                           "
64.                "                                                    "
64.                "ACCEPT'HASH'STORE PERFORMS THE FOLLOWING TEMPORAL SEQUENCE      "
64.                "       1.      READ DATA INTO POINTED LOCATION                  "
64.                "       2.      PERFORM CHECK ON XI                              "
64.                "       3.      LOCATE CODE FOR CURRENT CATEGORY                 "
64.                "       4.      CARRY OUT FUNCTION OF CODE FOR SPECIFIC CAT      "
64.                "       5.      EXIT,EXITS IF NOT VALID CATEGORY                 "
64.
64.
64.        MAIN:  READR(WORD[0]); "READ IN 15 DATA WORDS"
65.               XI=WORD[1];     "GET WORD 1"
66.               IF  XI <> 1;
67.                   "THEN"
67.                       BIT(WORD'16,1)=1;       "SET BIT 1"
68.               BIT(WORD'16,0)=1;       "SET BIT 0"
69.               IF  CAT = 'P';
70.                   "THEN"
70.
70.                       "WHEN CATEGORY IS EQUAL TO P, THE CODE "
70.                       "       1.  MAY CHECK VARIABLES SUB'CAT,FLT,SRN,MSN,SLOT'POINTER"
70.                       "       2.  WILL STORE FILTER'BITS"
70.                       "       3.  MAY OUTPUT ADDR'MSG"
70.                       "       4.  MAY SET BITS 10,11,14,15 OF WORD'16"
70.                       "       5.  WILL SET SLOT'POINTER"
70.
70.                       BEGIN   "CAT EQ P"
70.                       IF  SUB'CAT = 1;
71.                           "THEN"
71.                               BEGIN   "SUB'CAT EQ 1"
71.                               BIT(WORD'16,9)=1;       "SET BIT 9"
72.                               IF  FLT = LOR;
73.                                   "THEN"
73.                                       BIT(WORD'16,15)=1;      "SET BIT 15"
74.                               END     "SUB'CAT EQ 1"
75.                           ELSE
75.                               BEGIN   "SUB'CAT NE 1"
75.                               IF  SUB'CAT = 3;
77.                                   "THEN"
77.                                       BIT(WORD'16,10)=1;      "SET BIT 10"
78.                                   ELSE
79.                                       BIT(WORD'16,11)=1;      "SET BIT 11"
80.                               END     "SUB'CAT NE 1"
81.                       SRN=WORD[2];    "GET WORD 2"
82.                       GOTO BLK'8;
83.                       END     "CAT EQ P"
84.
84.
84.                   ELSE
84.                       BEGIN   "CAT NE P"
85.                       IF  CAT = 'T';
```

```
86.                                  "THEN"
86.
86.                                      "WHEN CATEGORY IS 1, THE CODE"
86.                                      "       1.  MAY CHECK SUB'CAT,ID,SRN,FIRST'LEVEL,MSN,SLOT'POINTER"
86.                                      "       2.  MAY OUTPUT SAM'AAA,ADDR'MSG"
86.                                      "       3.  WILL STORE FILTER'BITS"
86.                                      "       4.  MAY SET BITS 9,10,11,12,13,14 OF WORD'16"
86.                                      "       5.  WILL SET SLOT POINTER"
86.
86.                                      BEGIN   "CAT EQ T"
86.                                      IF  SUB'CAT = 1;
87.                                          "THEN"
87.                                              BEGIN   "SUB'CAT EQ 1"
87.                                              ID=WORD[9];     "GET WORD 9"
88.                                              IF  ID = 'F';
89.                                                  "THEN"
89.                                                      BIT(WORD'16,9)=1;       "SET BIT 9"
90.                                                  ELSE
91.                                                      BIT(WORD'16,12)=1;      "SET BIT 12"
92.                  T'WRD4:                             SRN=WORD[4];    "GET WORD 4"
93.                                              GOTO BLK'8;
94.                                              END     "SUB'CAT EQ 1"
95.                                          ELSE
95.                                              BEGIN   "SUB'CAT NE 1"
96.                                              IF  SUB'CAT = 5;
97.                                                  "THEN"
97.                                                      BEGIN   "SUB'CAT EQ 5"
97.                                                      ID=WORD[10];    "GET WORD 10"
98.                                                      IF  ID = 'F';
99.                                                          "THEN"
99.                                                              BEGIN   "ID EQ F"
99.                                                              BIT(WORD'16,11)=1;      "SET BIT 11"
100.                                                             GOTO T'WRD4;
101.                                                             END     "ID EQ F"
102.                                                         ELSE
102.                                                             BEGIN   "ID NE F"
103.                                                             IF  FIRST'LEVEL = 1;
104.                                                                 "THEN"
104.                                                                     BEGIN   "FIRST'LEVEL EQ 1"
104.                                                                     IF  SECOND'LEVEL = 3;
105.                                                                         "THEN"
105.                                                                             BEGIN   "SECOND'LEVEL EQ 3"
105.                                                                             OUTPUT(SAM'AAA);        "OUTPUT SAM'AAA"
106.                                                                             GOTO T'WRD4;
107.                                                                             END     "SECOND'LEVEL EQ 3"
108.                                                                     END     "FIRST'LEVEL EQ 1"
109.                                                                 ELSE
110.                                                                     GOTO BLK'10;
111.                                                             END     "ID NE F"
112.                                                     END     "SUB'CAT EQ 5"
113.                                                 ELSE
113.                                                     BEGIN   "SUB'CAT NE 5"
114.                                                     IF  SUB'CAT = 3;
```

```
115.                                                    "THEN"
115.                                                        ID=WORD[9];     "GET WORD 9"
116.                                                    ELSE
117.                                                        ID=WORD[10];    "GET WORD 10"
118.                                                IF  ID = 'F';
119.                                                    "THEN"
119.                                                        BIT(WORD'16,11)=1;      "SET BIT 11"
120.                                                    ELSE
121.                                                        BIT(WORD'16,13)=1;      "SET BIT 13"
122.                                                GOTO T'WRD4;
123.                                                END     "SUB'CAT NE 5"
124.                                            END     "SUB'CAT NE 1"
125.                                        END     "CAT EQ T"
126.
126.
126.                                    ELSE
126.                                        BEGIN   "CAT NE T"
127.                                        IF  CAT = 'I';
128.                                            "THEN"
128.
128.                                                "WHEN CATEGORY IS I, THE CODE"
128.                                                "       1. MAY CHECK VARIABLES SUB'CAT,LABEL,NON'ADDR'MSG,HOSTILE,SRN"
128.                                                "          ADDRESSE,MISSION,SLOT'POINTER                "
128.                                                "       2.  MAY STORE WEATHER,WEATHER'WARNING,NON'ADDR'MSG,SAM'AAA,ADDR'MSG"
128.                                                "       3.  MAY RESET SLOT'POINTER"
128.
128.                                                BEGIN   "CAT EQ I"
128.                                                IF  SUB'CAT = 1;
129.                                                    "THEN"
129.                                                        BEGIN   "SUB'CAT EQ 1"
129.                                                        IF  (LABEL = 1) OR (LABEL =  2);
130.                                                            "THEN"
130.                                                                WRITEW(WEATHER);        "STORE WEATHER"
131.                                                            ELSE
131.                                                                BEGIN   "LABEL NE 1 OR 2"
132.                                                                IF  LABEL = 3;
133.                                                                    "THEN"
133.                                                                        BEGIN   "LABEL EQ 3"
133.                                                                        IF  NON'ADDR'MSG = WANT;
134.                                                                            "THEN"
134.                                                                                WRITEW(NON'ADDR'MSG);   "STORE NON'ADDR'MSG"
135.                                                                            ELSE
136.                                                                                GOTO BLK'10;
137.                                                                        END     "LABEL EQ 3"
138.                                                                    ELSE
139.                                                                        WRITEW(WEATHER'WARNING);        "STORE WEATHER'WARNING"
140.                                                                END     "LABEL NE 1 OR 2"
141.                                                        END     "SUB'CAT EQ 1"
142.                                                    ELSE
142.                                                        BEGIN   "SUB'CAT NE 1"
143.                                                        IF  SUB'CAT = 3;
144.                                                            "THEN"
144.                                                                GOTO BLK'13;
```

```
145.                                                ELSE
145.                                                    BEGIN   "SUB'CAT NE 3"
146.                                                    IF  SUB'CAT = 7;
147.                                                        "THEN"
147.                                                            BEGIN   "SUB'CAT EQ 7"
147.                                                            IF  LABEL = 1;
148.                                                                "THEN"
148.                                                                    BEGIN   "LABEL EQ 1"
148.                                                                    HOSTILE=WORD[8];        "GET WORD 8"
149.                                                                    IF  HOSTILE = ON;
150.                                                                        "THEN"
150.                                                                            WRITEW(SAM'AAA);        "STORE SAM'AAA"
151.                                                                    SRN=WORD[4];    "GET WORD 4"
152.                                                                    IF  SRN = DEST'TGT;
153.                                                                        "THEN"
153.                                                                            WRITEW(ADDR'MSG);       "STORE ADDR'MSG"
154.                                                                    END     "LABEL EQ 1"
155.                                                                ELSE
156.                                                                    GOTO BLK'13;
157.                                                            END     "SUB'CAT EQ 7"
158.                                                        ELSE
159.                                                            GOTO BLK'10;
160.                                                    END     "SUB'CAT NE 3"
161.                                            END     "SUB'CAT NE 1"
162.                                    END     "CAT EQ I"
163.
163.                                ELSE
163.                                    BEGIN   "CAT NE I"
164.                                    IF  (CAT = 'V') OR (CAT = 'A') OR (CAT = 'M');
165.                                        "THEN"
165.
165.                                            "WHEN CATEGORY IS V, A, OR M, THE CODE"
165.                                            "       1.  MAY STORE ADDR'MSG          "
165.
165.                                            GOTO BLK'13;
166.
166.                                        ELSE
166.                                            BEGIN   "CAT NE V, A, OR M"
167.                                            IF  CAT = 'C';
168.                                                "THEN"
168.
168.                                                    "WHEN CATEGORY IS C, THE CODE   "
168.                                                    "       1.  MAY CHECK SUB'CAT,LABEL,NON'ADDR'MSG,ADDRESSE,MISSION"
168.                                                    "       2.  MAY STORE ADDR'MSG,NON'ADDR'MSG"
168.                                                    "       3.  MAY RESET SLOT'POINTER      "
168.
168.                                                    BEGIN   "CAT EQ C"
168.                                                    IF  (SUB'CAT = 2) AND ((LABEL = 2) OR (LABEL = 3) OR (LABEL =4) OR (
LABEL = 5));
169.                                                        "THEN"
169.                                                            BEGIN   "SUB'CAT EQ 2 AND LABEL EQ 2,3,4 OR 5"
169.                                                            IF  NON'ADDR'MSG = WANT;
170.                                                                "THEN"
170.                                                                    WRITEW(NON'ADDR'MSG);   "STORE NON'ADDR'MSG"
```

```
171.                                                                    ELSE
172.                                                                        GOTO BLK'10;
173.                                                                END     "SUB'CAT EQ 2 AND LABEL EQ 2,3,4 OR 5"
174.                                                            ELSE
175.                                                                GOTO BLK'13;
176.                                                        END     "CAT EQ C"
177.
177.                                                    ELSE
177.                                                        BEGIN   "CAT NE C"
178.                                                        IF  CAT = 'S';
179.                                                            "THEN"
179.                                                                "WHEN CATEGORY IS S, THE CODE"
179.                                                                "       1.  MAY STORE SENSOR MESSAGE"
179.
179.                                                                BEGIN   "CAT EQ S"
179.                                                                IF  SENSOR'MSG = WANT;
180.                                                                    "THEN"
180.                                                                        WRITEW(SENSOR'MSG);     "STORE SENSOR'MSG"
181.                                                                END     "CAT EQ S"
182.
182.                                                        END     "CAT NE C"
183.                                                END     "CAT NE V,A OR M"
184.                                            END     "CAT NE I"
185.                                        END     "CAT NE T"
186.                                END     "CAT NE P"
187.        OUT:    RETURN; "EXIT FROM ROUTINE"
188.                END     "ACCEPT'HASH'STORE"
```

STATISTIC NAME	OCCURRENCES	PERCENTAGE
CHARACTERS	7616	
LINES	343	
SYMBOLS	820	
KEY WORDS	196	23.90
AND	1	0.51
BEGIN	32	16.33
BIT	13	6.63
ELSE	26	13.27
END	32	16.33
GOTO	16	8.16
IF	35	17.86
ITEM	32	16.33
OR	6	3.6
PROC	1	0.51
RETURN	1	0.51
TABLE	1	0.51
COMMENTS	192	23.41
DIRECTIVES	4	0.49
COMPOOL	1	25.0
EJECT	3	75.0
CONSTANTS	80	9.76
INTEGER	68	85.0
CHARACTER	12	15.0
SIGNS	331	40.37
+	1	0.30
=	68	20.54
<>	1	0.30
,	14	4.23
:	7	2.11
;	130	39.27
!	4	1.21
(	38	11.48
)	38	11.48
[	15	4.53
]	15	4.53
ABBREVIATIONS/DEFINE FORMAL PARAMETERS	32	3.90
C	2	6.25
U	30	93.75
NAMES	180	21.95
COMPOOL	1	0.56
DEFINE	1	0.56
LABEL	22	12.22
PROC	14	7.78
TABLE	1	0.56
TABLE-ITEM	14	7.78
SIMPLE-ITEM	124	68.89
DECLARATIONS	34	
SIMPLE-ITEM	31	91.18
NON-BASED	31	100.0
TABLE	1	2.94

STATISTIC NAME	OCCURRENCES	PERCENTAGE
TABLE-ITEM	1	2.94
PROC	1	2.94
STATEMENTS	189	
SINGLE ASSIGNMENT	27	14.29
PROC CALL	13	6.88
IF	35	18.52
GOTO	16	8.47
RETURN	1	0.53

```
STMT    LOC R    CODE R  CODE R   LABEL     MNEM  OPERANDS
                                              ADDR'MSG
        01A4 H   0028 L                       FILTER'BIT
        01A5 H   0027 L                       WORD-1
        01A6 H   0000 L                       SAM'AAA
        01A7 H   0026 L                       WEATHER
        01A8 H   0029 L                       NON'ADDR'M
        01A9 H   0018 L                       WEATHER'WA
        01AA H   002A L                       SENSOR'MSG
        01AB H   0022 L
 . REGION 1 *
2.      0000 H                    ACCEPT'
 . REGION 2 *
36.     0002 H   70F0   0050 H                J       MAIN
38.     0004 H                    BLK'8
 . REGION 3 *
        0004 H   8000   0014 L                L       0,SRN
        0006 H   F000   0020 L                C       0,DEST
        0008 H   7050   0012 H                JC      NE,H.12
39.     000A H   83F0   01A4 H                LIM     15,(ADDR'MSG)
        000C H   7220   0000 *                JS      2,OUTPUT
40.     000E H   50E0   0000 L                SB      14,WORD'16
42.     0010 H   70F0   001C H                J       H.1C
43.     0012 H   8010   000F L     H.12       L       1,WORD+14
        0014 H   9010   001C L                ST      1,MSN
44.     0015 H   F010   0023 L                C       1,OWN'MSN
        0018 H   7050   001C H                JC      NE,H.1C
45.     001A H   50E0   0000 L                SB      14,WORD'16
47.     001C H   83F0   01A5 H     H.1C       LIM     15,(FILTER'BIT)
        001E H   7220   0000 *                JS      2,WRITEW
49.     0020 H                    BLK'10
 . REGION 4 *
        0020 H   8000   002B L                L       0,SLOT'POINT
        0022 H   F000   002C L                C       0,END'POINTE
        0024 H   7050   002C H                JC      NE,H.2C
50.     0026 H   8000   002D L                L       0,TOP'POINTE
        0028 H   9000   002B L                ST      0,SLOT'POINT
51.     002A H   70F0   0032 H                J       H.32
52.     002C H   8300   000F     H.2C        LIM     0,15
        002E H   A000   002B L                A       0,SLOT'POINT
        0030 H   9000   002B L                ST      0,SLOT'POINT
53.     0032 H   70F0   01A0 H     H.32       J       OUT
54.     0034 H                    BLK'13
 . REGION 5 *
        0034 H   8010   0004 L                L       1,WORD+3
        0036 H   9010   0019 L                ST      1,ADDRESSE
55.     0038 H   F010   0024 L                C       1,SELF
        003A H   7050   0042 H                JC      NE,H.42
56.     003C H   83F0   01A4 H                LIM     15,(ADDR'MSG)
        003E H   7220   0000 *                JS      2,WRITEW
57.     0040 H   70F0   004E H                J       H.4E
58.     0042 H   8010   000F L     H.42       L       1,WORD+14
        0044 H   9010   001D L                ST      1,MISSION
```

```
STMT    LOCN R    CODE R   CODE R   LABEL    MNEM   OPERANDS
59.     0046 H    F010     0025 L            C      1,OWN
        0048 H    7050     004E H            JC     NE,H.4E
60.     004A H    83F0     01A4 H            LIM    15,(ADDR'MSG)
        004C H    7220     0000 *            JS     2,WRITEW
62.     004E H    70F0     01A0 H   H.4E     J      OUT
64.     0050 H                      MAIN
. REGION 6 *
        0050 H    83F0     01A6 H            LIM    15,(WORD-1)
        0052 H    7220     0000 *            JS     2,READR
65.     0054 H    8010     0001 L            L      1,WORD
        0056 H    9010     0010 L            ST     1,XI
66.     0058 H    F310     0001              CIM    1,\1
        005A H    7020     005E H            JC     EQ,H.5E
67.     005C H    5010     0000 L            SB     1,WORD'16
68.     005E H    5000     0000 L   H.5E      SB     0,WORD'16
69.     0060 H    8010     0011 L            L      1,CAT
        0062 H    F310     5020              CIM    1,'P '
        0064 H    7050     0088 H            JC     NE,H.88
70.     0066 H    8020     0012 L            L      2,SUB'CAT
        0068 H    F320     0001              CIM    2,\1
        006A H    7050     0078 H            JC     NE,H.78
71.     006C H    5090     0000 L            SB     9,WORD'16
72.     006E H    8000     0013 L            L      0,FLT
        0070 H    F000     001E L            C      0,LDR
        0072 H    7050     0076 H            JC     NE,H.76
73.     0074 H    50F0     0000 L            SB     15,WORD'16
75.     0076 H    70F0     0082 H   H.76      J      H.82
76.     0078 H    F320     0003     H.78     CIM    2,\3
        007A H    7050     0080 H            JC     NE,H.80
77.     007C H    50A0     0000 L            SB     10,WORD'16
78.     007E H    70F0     0082 H            J      H.82
79.     0080 H    50B0     0000 L   H.80      SB     11,WORD'16
81.     0082 H    8000     0002 L   H.82      L      0,WORD+1
        0084 H    9000     0014 L            ST     0,SRN
82.     0086 H    70F0     0004 H            J      BLK'8
85.     0088 H    F310     5420     H.88     CIM    1,'T '
        008A H    7050     00EE H            JC     NE,H.EE
86.     008C H    8000     0012 L            L      0,SUB'CAT
        008E H    F300     0001              CIM    0,\1
        0090 H    7050     00A6 H            JC     NE,H.A6
87.     0092 H    8010     0009 L            L      1,WORD+8
        0094 H    9010     0015 L            ST     1,ID
88.     0096 H    F310     0046              CIM    1,70
        0098 H    7050     009E H            JC     NE,H.9E
89.     009A H    5090     0000 L            SB     9,WORD'16
90.     009C H    70F0     00A0 H            J      H.A0
91.     009E H    50C0     0000 L   H.9E      SB     12,WORD'16
92.     00A0 H                      H.A0
        00A0 H                      T'WRD4
. REGION 7 *
        00A0 H    8000     0004 L            L      0,WORD+3
        00A2 H    9000     0014 L            ST     0,SRN
```

```
STMT   LOCN R    CODE R   CODE R    LABEL    MNEM   OPERANDS
93.    00A4 H    70F0     0004 R               J      BLK'8
95.    00A6 H                        H.A6
 . REGION 8 .
96.    00A6 H    8010     0012 L               L      1,SUB'CAT
       00A8 H    F310     0005                 CIM    1,\5
       00AA H    7050     00D0 H               JC     NE,H.D0
97.    00AC H    8020     000A L               L      2,WORD+9
       00AE H    9020     0015 L               ST     2,ID
98.    00B0 H    F320     0046                 CIM    2,70
       00B2 H    7050     00B8 H               JC     NE,H.B8
99.    00B4 H    5080     0000 L               SB     11,WORD'16
100.   00B6 H    70F0     00A0 H               J      T'WRD4
103.   00B8 H    8000     0016 L      H.B8     L      0,FIRST'LEVE
       00BA H    F300     0001                 CIM    0,\1
       00BC H    7050     00CC H               JC     NE,H.CC
104.   00BE H    8000     0017 L               L      0,SECOND'LEV
       00C0 H    F300     0003                 CIM    0,\3
       00C2 H    7050     00CA H               JC     NE,H.CA
105.   00C4 H    83F0     01A7 H               LIM    15,(SAM'AAA)
       00C6 H    7220     0000 *               JS     2,OUTPUT
106.   00C8 H    70F0     00A0 H               J      T'WRD4
109.   00CA H    70F0     00CE H      H.CA     J      H.CE
110.   00CC H    70F0     0020 H      H.CC     J      BLK'10
113.   00CE H    70F0     00EC H      H.CE     J      H.EC
114.   00D0 H    F310     0003        H.D0     CIM    1,\3
       00D2 H    7050     00DA H               JC     NE,H.DA
115.   00D4 H    8000     0009 L               L      0,WORD+8
       00D6 H    9000     0015 L               ST     0,ID
116.   00D8 H    70F0     00DE H               J      H.DE
117.   00DA H    8000     000A L      H.DA     L      0,WORD+9
       00DC H    9000     0015 L               ST     0,ID
118.   00DE H    8000     0015 L      H.DE     L      0,ID
       00E0 H    F300     0046                 CIM    0,70
       00E2 H    7050     00E8 H               JC     NE,H.E8
119.   00E4 H    5080     0000 L               SB     11,WORD'16
120.   00E6 H    70F0     00EA H               J      H.EA
121.   00E8 H    5000     0000 L      H.E8     SB     13,WORD'16
122.   00EA H    70F0     00A0 H      H.EA     J      T'WRD4
126.   00EC H    70F0     01A0 H      H.EC     J      H.1A0
       00EE H                        H.EE
 . REGION 9 .
127.   00EE H    8010     0011 L               L      1,CAT
       00F0 H    F310     4920                 CIM    1,'I '
       00F2 H    7050     0154 H               JC     NE,H.154
128.   00F4 H    8020     0012 L               L      2,SUB'CAT
       00F6 H    F320     0001                 CIM    2,\1
       00F8 H    7050     0124 H               JC     NE,H.124
129.   00FA H    8030     001A L               L      3,LABEL
       00FC H    F330     0001                 CIM    3,\1
       00FE H    7020     0104 H               JC     EQ,H.104
       0100 H    F330     0002                 CIM    3,\2
       0102 H    7050     010A H               JC     NE,H.10A
```

```
STMT   LOC  R   CODE  R  CODE  R   LABEL     MNEM   OPERANDS
130.   0104 H   83F0      01AB  H   H.104     LIM    15,(WEATHER)
       0106 H   722Ø      0000  *               JS     2,WRITEW
131.   0108 H   70F0      0122  H               J      H.122
132.   010A H   F330      0003        H.10A     CIM    3,\3
       010C H   7050      011E  H               JC     NE,H.11E
133.   010E H   8030      0018  L               L      3,NON'ADDR'M
       0110 H   F030      001F  L               C      3,WANT
       0112 H   7050      011A  H               JC     NE,H.11A
134.   0114 H   83F0      01A9  H               LIM    15,(NON'ADDR'M)
       0116 H   722Ø      0000  *               JS     2,WRITEW
135.   0118 H   70F0      011C  H               J      H.11C
136.   011A H   70F0      0020  H   H.11A       J      BLK'1Ø
138.   011C H   70F0      0122  H   H.11C       J      H.122
139.   011E H   83F0      01AA  H   H.11E       LIM    15,(WEATHER'WA)
       0120 H   722Ø      0000  *               JS     2,WRITEW
142.   0122 H   70F0      0152  H   H.122       J      H.152
143.   0124 H   F320      0003        H.124      CIM    2,\3
       0126 H   7020      0034  H               JC     EQ,BLK'13
146.   0128 H   F320      0007        H.128      CIM    2,\7
       012A H   7050      0150  H               JC     NE,H.150
147.   012C H   8320      0001                  LIM    2,\1
       012E H   F020      001A  L               C      2,LABEL
       0130 H   7050      014C  H               JC     NE,H.14C
148.   0132 H   8030      0008  L               L      3,WORD+7
       0134 H   9030      0019  L               S1     3,HOSTILE
149.   0136 H   F330      0001                  CIM    3,\1
       0138 H   7050      013E  H               JC     NE,H.13E
150.   013A H   83F0      01A7  H               LIM    15,(SAM'AAA)
       013C H   722Ø      0000  *               JS     2,WRITEW
151.   013E H   8010      0004  L   H.13E       L      1,WORD+3
       0140 H   9010      0014  L               ST     1,SRN
152.   0142 H   F010      0021  L               C      1,DEST'TGT
       0144 H   7050      014A  H               JC     NE,H.14A
153.   0146 H   83F0      01A4  H               LIM    15,(ADDR'MSG)
       0148 H   722Ø      0000  *               JS     2,WRITEW
155.   014A H   70F0      014E  H   H.14A       J      H.14E
156.   014C H   70F0      0034  H   H.14C       J      BLK'13
158.   014E H   70F0      0152  H   H.14E       J      H.152
159.   0150 H   70F0      0020  H   H.150       J      BLK'1Ø
163.   0152 H   70F0      01A0  H   H.152       J      H.1A0
164.   0154 H   F310      5620        H.154      CIM    1,'V '
       0156 H   7020      0160  H               JC     EQ,H.160
       0158 H   F310      4120                  CIM    1,'A '
       015A H   7020      0160  H               JC     EQ,H.160
       015C H   F310      4D20                  CIM    1,'M '
       015E H   7050      0162  H               JC     NE,H.162
165.   0160 H   70F0      0034  H   H.160       J      BLK'13
167.   0162 H   F310      4320        H.162      CIM    1,'C '
       0164 H   7050      0192  H               JC     NE,H.192
168.   0166 H   8320      0002                  LIM    2,\2
       0168 H   F020      0012  L               C      2,SUB'CAT
       016A H   7050      018E  H               JC     NE,H.18E
```

```
STMT    LOCN R    CODE R   CODE R    LABEL     MNEM   OPERANDS
        016C H    8030     0001A L             L      3,LABEL
        016E H    F330     0002               CIM    3,\2
        0170 H    7020     017E H             JC     EQ,H.17E
        0172 H    F330     0003               CIM    3,\3
        0174 H    7020     017E H             JC     EQ,H.17E
        0176 H    F330     0004               CIM    3,\4
        0178 H    7020     017E H             JC     EQ,H.17E
        017A H    F330     0005               CIM    3,\5
        017C H    7050     018E H             JC     NE,H.18E
169.    017E H    8010     0018 L    H.17E    L      1,NON'ADDR'M
        0180 H    F010     001F L             C      1,WANT
        0182 H    7050     018A H             JC     NE,H.18A
170.    0184 H    83F0     01A9 H             LIM    15,(NON'ADDR'M)
        0186 H    7220     0000 *             JS     2,WRITEW
171.    0188 H    70F0     018C H             J      H.18C
172.    018A H    70F0     0020 H    H.18A    J      BLK'10
174.    018C H    70F0     0190 H    H.18C    J      H.190
175.    018E H    70F0     0034 H    H.18E    J      BLK'13
177.    0190 H    70F0     01A0 H    H.190    J      H.1A0
178.    0192 H    F310     5320      H.192    CIM    1,'S '
        0194 H    7050     01A0 H             JC     NE,H.1A0
179.    0196 H    8010     0022 L             L      1,SENSOR'MSG
        0198 H    F010     001F L             C      1,WANT
        019A H    7050     01A0 H             JC     NE,H.1A0
180.    019C H    83F0     01AB H             LIM    15,(SENSOR'MSG)
        019E H    7220     0000 *             JS     2,WRITEW
185.    01A0 H                       H.1A0
. REGION 10*
187.    01A0 H                       H.1A0
        01A0 H                       OUT
. REGION 11*
188.    01A0 H    8F30     0030 L    H.1A0    LM     3,L.30
        01A2 H    70F2     0000               J      \0,2

        0000 H    9F30     0030 L             STM    3,L.30
. REGION 12*
189.                                         END
```

NAME	CLAS	SCOPE	LOC	TY	FB	SIZ	PERM/DSIZ	DEF	- - SET(*)/USED - -								
ACCEPT'HAS	PROC	*GLOBAL*	0000H				7	2:									
ADDR'MSG	ITEM	ACCEPT'HAS	0028L	U	0	16	RESERV	30:	39	56	60	153					
ADDRESSE	ITEM	ACCEPT'HAS	0019L	U	0	16	RESERV	15:	54*	55							
ATAN	PROC	*GLOBAL*	0000	F	0	32	0	2:									
BLK'10	LABL	ACCEPT'HAS	0020H					49:	110	136	159	172					
BLK'13	LABL	ACCEPT'HAS	0034H					54:	144	156	165	175					
BLK'8	LABL	ACCEPT'HAS	0004H					38:	82	93							
CAT	ITEM	ACCEPT'HAS	0011L	C	0	2	RESERV	7:	69	85	127	164	164	164	167	178	
COS	PROC	*GLOBAL*	0000	F	0	32	0	2:									
DATA	TABL	ACCEPT'HAS	0001L	U	0	16	RESERV	4:									
DEST	ITEM	ACCEPT'HAS	0024L	U	0	16	RESERV	22:	38								
DEST'TGT	ITEM	ACCEPT'HAS	0021L	U	0	16	RESERV	23:	152								
END'POINTE	ITEM	ACCEPT'HAS	0026L	U	0	16	RESERV	34:	49								
FILTER'BIT	ITEM	ACCEPT'HAS	0027L	U	0	16	RESERV	29:	47								
FIRST'LEVE	ITEM	ACCEPT'HAS	0015L	U	0	16	RESERV	12:	103								
FLT	ITEM	ACCEPT'HAS	0013L	U	0	16	RESERV	9:	72								
HOSTILE	ITEM	ACCEPT'HAS	0018L	U	0	16	RESERV	17:	148*	149							
ID	ITEM	ACCEPT'HAS	0015L	U	0	16	RESERV	11:	87*	88	97*	98	115*	117*	118		
JTIDS	BLOC	*GLOBAL*	0000*			0	EXTRNL	2:									
LABEL	ITEM	ACCEPT'HAS	001AL	U	0	16	RESERV	16:	129	129	132	147	168	168	168	168	
LDR	ITEM	ACCEPT'HAS	001EL	U	0	16	RESERV	20:	72								
MAIN	LABL	ACCEPT'HAS	0050H					64:	36								
MISSION	ITEM	ACCEPT'HAS	001DL	U	0	16	RESERV	19:	58*	59							
MSN	ITEM	ACCEPT'HAS	001CL	U	0	16	RESERV	18:	43*	44							
NON'ADDR'M	ITEM	ACCEPT'HAS	0018L	U	0	16	RESERV	14:	133	134	169	170					
ON	DEFN	*GLOBAL*						2:	149								
OUT	LABL	ACCEPT'HAS	01A0H					187:	53	62							
OUTPUT	PROC	*GLOBAL*	0000*				0	2:	39	105							
OWN	ITEM	ACCEPT'HAS	0025L	U	0	16	RESERV	27:	59								
OWN'MSN	ITEM	ACCEPT'HAS	0023L	U	0	16	RESERV	25:	44								
PERFORM	DEFN	*GLOBAL*						2:									
READP	PROC	*GLOBAL*	0000*				0	2:	64								
SAM'AAA	ITEM	ACCEPT'HAS	0026L	U	0	16	RESERV	28:	105	150							
SECOND'LEV	ITEM	ACCEPT'HAS	0017L	U	0	16	RESERV	13:	104								
SELF	ITEM	ACCEPT'HAS	0024L	U	0	16	RESERV	26:	55								
SENSOR'MSG	ITEM	ACCEPT'HAS	0022L	U	0	16	RESERV	24:	179	180							
SIN	PROC	*GLOBAL*	0000	F	0	32	0	2:									
SLOT'POINT	ITEM	ACCEPT'HAS	0028L	U	0	16	RESERV	33:	49	50*	52	52*					
SQRT	PROC	*GLOBAL*	0000	F	0	32	0	2:									
SRN	ITEM	ACCEPT'HAS	0014L	U	0	16	RESERV	10:	38	81*	92*	151*	152				
SUB'CAT	ITEM	ACCEPT'HAS	0012L	U	0	16	RESERV	8:	70	76	86	96	114	128	143	146	168
T'WPD4	LABL	ACCEPT'HAS	00A0H					92:	100	106	122						
TAN	PROC	*GLOBAL*	0000	F	0	32	0	2:									
TOP'POINTE	ITEM	ACCEPT'HAS	0020L	U	0	16	RESERV	35:	50								
WANT	ITEM	ACCEPT'HAS	001FL	U	0	16	RESERV	21:	133	169	179						
WEATHER	ITEM	ACCEPT'HAS	0029L	U	0	16	RESERV	31:	130								
WEATHER'WA	ITEM	ACCEPT'HAS	002AL	U	0	16	RESERV	32:	139								
WORD	ITEM	DATA	0001L	C	0	2	RESERV	5:	43	54	58	64	65	81	87	92	97
									115	117	148	151					
WORD'16	ITEM	ACCEPT'HAS	0000L	U	0	16	RESERV	3:	40*	45*	67*	68*	71*	73*	77*	79*	89*
									91*	99*	119*	121*					

```
JOVIAL  V.062876  9/14/76 14:39  MODULE:MBLT.ASM      PAGE 17
NAME        CLAS SCOPE      LOC      TY FB   SIZ  PERM/OSIZ DEF    - - SET(*)/USED - -

WRITEW      PROC *GLOBAL*   0000                  0          2:   47      56      60      130     134     139     150     153     170
                                                                                180
XI          ITEM ACCEPT*HAS 0010L    U  0     16  RESERV     6:   65*     66

PROGRAM SUMMARY
DATA/VARIABLES            0000 - 0035
INSTRUCTIONS/CONSTANTS    8000 - 81AB
EXTERNS: WRITEW READR  OUTPUT JTIDS
INTERNS: ACCEPT
FILES REFERENCED:
   JTIDS.CMP                 9/14/76   9:36   CMP:JTIDS

343 LINES  91 MESSAGES:   91 INFORMATION
CPU TIME     9.319 SEC
```

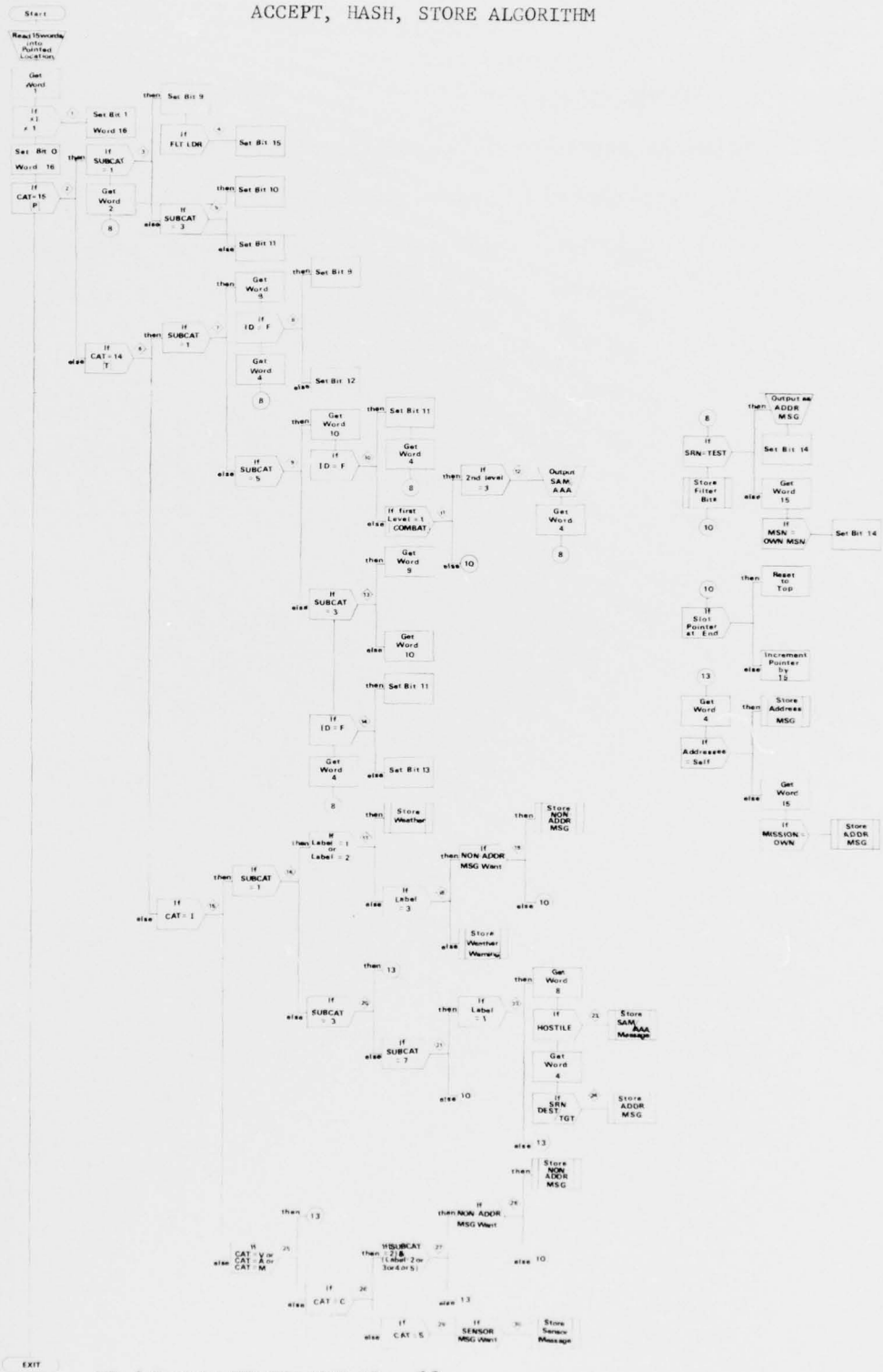
APPENDIX C

ALGORITHM #2, ACCEPT/HASH/STORE ("STRUCTURED" FLOW CHART)

ALGORITHM #2 (ACCEPT/HASH/STORE) DESCRIPTION

This section documents the results of restructuring the logic structure contained in Appendix "B". In lieu of the logic conventions of Appendix "B" (FORTRAN-type IF statements, GO TO's, etc), a set of structured programming control constructs (IF THEN ELSE, FOR WHILE, etc) were used. The algorithm (ACCEPT/HASH/STORE) was re-charted and in turn re-coded in JOVIAL-73/I. The resultant flow chart is contained on the following page.

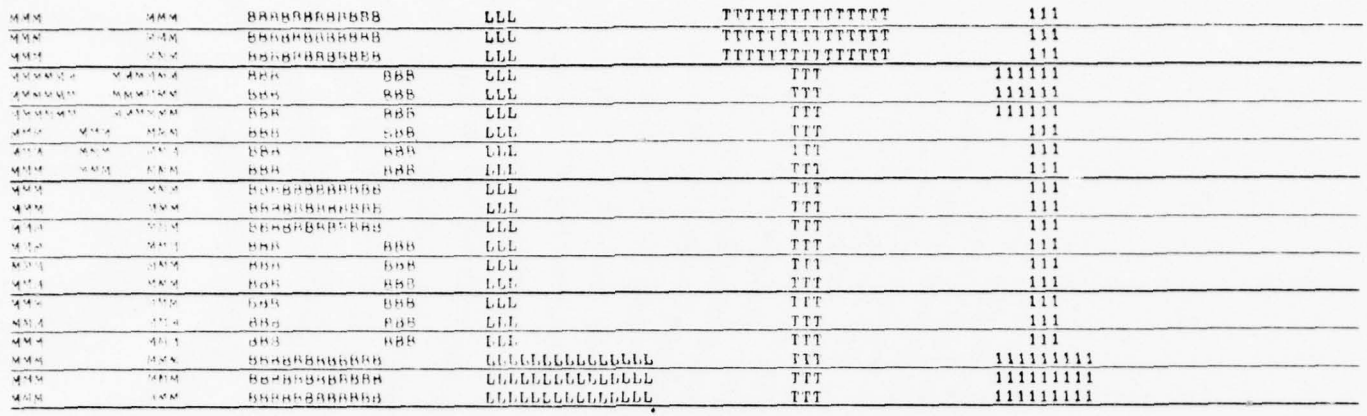
ACCEPT, HASH, STORE ALGORITHM

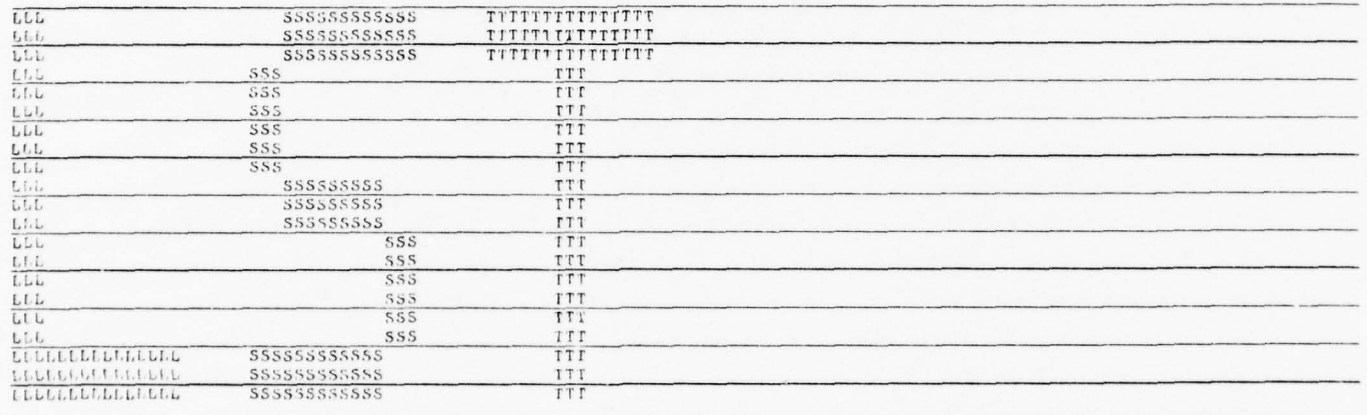


NOTE * Corresponds to CONDn label in computer code

JOVIAL-73/I IMPLEMENTATION

The following pages are the resultant compiler output for the JOVIAL-73/I compilation of Algorithm #2, ACCEPT/HASH/STORE. This procedure (named MBLT 1) is coded according to structured programming guidelines.






```
LPTSPL Version 6(347)  Running on OPTM11
 *START* User MURLAROFF [32,2,1376] Job  MBLT1 Seq. 543 Date 14-Sep-76 12:58:16 Monitor AFAL 6M2.15 SYSTEM *START*
Request created: 14-Sep-76  12:59:08
File: DSKH1:MBLT1.LST[32,2,1376] Created: 14-Sep-76 12:57:00 Printed: 14-Sep-76 12:58:30
QUEUE Switches:  /PRINTER:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:112 /FORMS:NORMAL
File will be deleted after printing
```

MBLT1,MBLT1=MBLT1.JTS/HBC/MAC/ACR/NOIN/STAT

```
1.          !COMPOOL('JTIDS.CMP');
2.          PROC    ACCEPT'HASH'STORE;
3.          "MAIN PROGRAM BELOW NESTED PROCS"
3.                  BEGIN   "ACCEPT'HASH'STORE"
3.
3.                  "DECLARATION SECTION"
3.
3.                  "EXTERNAL DECLARATIONS"
3.
3.                  ITEM    WORD'16 U;
4.
4.                  "LOCAL DECLARATIONS"
4.
4.                  TABLE   DATA(1:15)      1;
5.                          ITEM    WORD    C 2(0,0);
6.                  ITEM    XI      U;
7.                  ITEM    CAT     C 2;
8.                  ITEM    SUB'CAT U;
9.                  ITEM    FLT     U;
10.                 ITEM    SRN     U;
11.                 ITEM    ID      U;
12.                 ITEM    FIRST'LEVEL     U;
13.                 ITEM    SECOND'LEVEL    U;
14.                 ITEM    NON'ADDR'MSG    U;
15.                 ITEM    ADDRESSE        U;
16.                 ITEM    LABEL   U;
17.                 ITEM    HOSTILE U;
18.                 ITEM    MSN     U;
19.                 ITEM    MISSION U;
20.
21.                 ITEM    LDR     U;
21.                 ITEM    WANT    U;
22.                 ITEM    DEST    U;
23.                 ITEM    DEST'TGT        U;
24.                 ITEM    SENSOR'MSG      U;
25.                 ITEM    OWN'MSN U;
26.                 ITEM    SELF    U;
27.                 ITEM    OWN     U;
28.
28.                 ITEM    SAM'AAA U;
29.                 ITEM    FILTER'BITS     U;
30.                 ITEM    ADDR'MSG        U;
31.                 ITEM    WEATHER U;
32.                 ITEM    WEATHER'WARNING U;
33.
33.                 ITEM    SLOT'POINTER    U;
34.                 ITEM    END'POINTER     U;
35.                 ITEM    TOP'POINTER     U;
36.
36.         !EJECT;
```

```
37.                     PROC    BLK'10;
38.                     "                                       "
38.                     "THIS PROCEDURE RESETS SLOT POINTER     "
38.                     "TO EITHER 15 MORE OR TO TOP            "
39.                     "                                       "
38.
38.                             BEGIN   "BLK'10"
38.                             IF  SLOT'POINTER = END'POINTER;
39.                                 "THEN"
39.                                     SLOT'POINTER=TOP'POINTER;       "RESET TO TOP"
40.                                 ELSE
41.                                     SLOT'POINTER=SLOT'POINTER+15;   "ADD 15"
42.                             RETURN;
43.                             END     "BLK'10"
44.
44.             !EJECT;
```

```
45.                     PROC    BLK'8;
45.                     "                                                       "
46.                     "THIS PROCEDURE IS COMMON BETWEEN CATEGORY P AND T      "
46.                     "IT CHECKS IF SRN EQ DEST, IF SO OUTPUTS ADDR'MSG AND SETS BIT 14      "
46.                     "  IF NOT, GETS WORD 15, CHECKS IF OWN'MSN,  IF SO, SETS BIT 14"
46.                     "IT STORES FILTER'BITS AND PERFORMS BLK'10                    "
46.                     "                                                       "
46.
46.                         BEGIN   "BLK'8"
46.                         IF  SRN = DEST;
47.                             "THEN"
47.                                 BEGIN   "SRN EQ DEST"
47.                                 OUTPUT(ADDR'MSG);       "OUTPUT ADDR'MSG"
48.                                 BIT(WORD'16,14)=1;      "SET BIT 14"
49.                                 END     "SRN EQ DEST"
50.                             ELSE
50.                                 BEGIN   "SRN NE DEST"
51.                                 MSN=WORD[15];   "GET WORD 15"
52.                                 IF  MSN = OWN'MSN;
53.                                     "THEN"
53.                                         BIT(WORD'16,14)=1;      "SET BIT 14"
54.                                 END     "SRN NE DEST"
55.                         WRITEW(FILTER'BITS);    "STORE FILTER BITS"
56.                         PERFORM BLK'10;
57.                         RETURN;
58.                         END     "BLK'8"
59.
59.             !EJECT;
```

```
60.                     PROC    BLK'13;
61.                     "                                                                    "
61.                     "THIS PROCEDURE IS COMMON BETWEEN CATEGORIES V, A, M, I, AND C       "
61.                     "IT GETS WORD 4, THEN CHECKS ADDRESSE, IF SELF, STORES ADDR'MSG,      "
61.                     "  IF NOT SELF, GETS WORD 15 THEN CHECKS MISSION, IF OWN IT STORES ADDR'MSG"
61.                     "                                                                    "
61.
61.                             BEGIN   "BLK'13"
61.                             ADDRESSE=WORD[4];       "GET WORD 4"
62.                             IF  ADDRESSE = SELF;
63.                                 "THEN"
63.                                     WRITEW(ADDR'MSG);       "STORE ADDR'MSG"
64.                                 ELSE
64.                                     BEGIN   "ADDRESSE NE SELF"
65.                                     MISSION=WORD[15];       "GET WORD 15"
66.                                     IF  MISSION = OWN;
67.                                         "THEN"
67.                                             WRITEW(ADDR'MSG);       "STORE ADDR'MSG"
68.                                     END     "ADDRESSE NE SELF"
69.                             RETURN;
70.                             END     "BLK'13"
71.
71.             !EJECT;
```

```
72.
72.                    "                                                    "
72.                    "    ENTRY TO MAIN PROGRAM                           "
72.                    "                                                    "
72.                    "ACCEPT'HASH'STORE PERFORMS THE FOLLOWING TEMPORAL SEQUENCE      "
72.                    "       1.      READ DATA INTO POINTED LOCATION                  "
72.                    "       2.      PERFORM CHECK ON XI                              "
72.                    "       3.      LOCATE CODE FOR CURRENT CATEGORY                 "
72.                    "       4.      CARRY OUT FUNCTION OF CODE FOR SPECIFIC CAT      "
72.                    "       5.      EXIT,EXITS IF NOT VALID CATEGORY                 "
72.
72.
72.                 READR(WORD[0]); "READ IN 15 DATA WORDS"
73.                 XI=WORD[1];             "GET WORD 1"
74.          COND1: IF  XI <> 1;
75.                     "THEN"
75.                         BIT(WORD'16,1)=1;       "SET BIT 1"
76.                 BIT(WORD'16,0)=1;       "SET BIT 0"
77.          COND2: IF  CAT = 'P';
78.                     "THEN"
78.
78.                         "WHEN CATEGORY IS EQUAL TO P, THE CODE "
78.                         "       1.  MAY CHECK VARIABLES SUB'CAT,FLT,SRN,MSN,SLOT'POINTER"
78.                         "       2.  WILL STORE FILTER'BITS"
78.                         "       3.  MAY'OUTPUT ADDR'MSG"
78.                         "       4.  MAY SET BITS 10,11,14,15 OF WORD'16"
78.                         "       5.  WILL SET SLOT'POINTER"
78.
78.                         BEGIN   "CAT EQ P"
78.          COND3:         IF  SUB'CAT = 1;
79.                             "THEN"
79.                                 BEGIN   "SUB'CAT EQ 1"
79.                                 BIT(WORD'16,9)=1;       "SET BIT 9"
80.          COND4:                 IF  FLT = LOR;
81.                                     "THEN"
81.                                         BIT(WORD'16,15)=1;      "SET BIT 15"
82.                                 END     "SUB'CAT EQ 1"
83.                             ELSE
83.                                 BEGIN   "SUB'CAT NE 1"
84.          COND5:                 IF  SUB'CAT = 3;
85.                                     "THEN"
85.                                         BIT(WORD'16,10)=1;      "SET BIT 10"
86.                                     ELSE
87.                                         BIT(WORD'16,11)=1;      "SET BIT 11"
88.                                 END     "SUB'CAT NE 1"
89.                         SRN=WORD[2];    "GET WORD 2"
90.                         PERFORM BLK'8;
91.                         END     "CAT EQ P"
92.
92.
92.                     ELSE
92.                         BEGIN   "CAT NE P"
93.          COND6:         IF  CAT = 'T';
```

```
 94.                                "THEN"
 94.
 94.                                    "WHEN CATEGORY IS T, THE CODE"
 94.                                    "       1.  MAY CHECK SUB'CAT,ID,SRN,FIRST'LEVEL,MSN,SLOT'POINTER"
 94.                                    "       2.  MAY OUTPUT SAM'AAA,ADDR'MSG"
 94.                                    "       3.  WILL STORE FILTER'BITS"
 94.                                    "       4.  MAY SET BITS 9,10,11,12,13,14 OF WORD'16"
 94.                                    "       5.  WILL SET SLOT POINTER"
 94.
 94.                                    BEGIN   "CAT EQ T"
 94.          COND7:                    IF  SUB'CAT = 1;
 95.                                        "THEN"
 95.                                            BEGIN   "SUB'CAT EQ 1"
 95.                                            ID=WORD[9];     "GET WORD 9"
 96.          COND8:                            IF  ID = 'F';
 97.                                                "THEN"
 97.                                                    BIT(WORD'16,9)=1;       "SET BIT 9"
 98.                                                ELSE
 99.                                                    BIT(WORD'16,12)=1;      "SET BIT 12"
100.                                            SRN=WORD[4];    "GET WORD 4"
101.                                            PERFORM BLK'8;
102.                                            END     "SUB'CAT EQ 1"
103.                                        ELSE
103.                                            BEGIN   "SUB'CAT NE 1"
104.          COND9:                            IF  SUB'CAT = 5;
105.                                                "THEN"
105.                                                    BEGIN   "SUB'CAT EQ 5"
106.          COND10:                                   ID=WORD[10];    "GET WORD 10"
107.                                                    IF  ID = 'F';
107.                                                        "THEN"
107.                                                            BEGIN   "ID EQ F"
108.                                                            BIT(WORD'16,11)=1;      "SET BIT 11"
109.                                                            SRN=WORD[4];    "GET WORD 4"
110.                                                            PERFORM BLK'8;
111.                                                            END     "ID EQ F"
111.                                                        ELSE
112.          COND11:                                           BEGIN   "ID NE F"
113.                                                            IF  FIRST'LEVEL = 1;
113.                                                                "THEN"
113.          COND12:                                               BEGIN   "FIRST'LEVEL EQ 1"
114.                                                                IF  SECOND'LEVEL = 3;
114.                                                                    "THEN"
114.                                                                        BEGIN   "SECOND'LEVEL EQ 3"
114.                                                                        OUTPUT(SAM'AAA);        "OUTPUT SAM'AAA"
115.                                                                        SRN=WORD[4];    "GET WORD 4"
116.                                                                        PERFORM BLK'8;
117.                                                                        END     "SECOND'LEVEL EQ 3"
118.                                                                END     "FIRST'LEVEL EQ 1"
119.                                                            ELSE
120.                                                                PERFORM BLK'10;
121.                                                            END     "ID NE F"
122.                                                    END     "SUB'CAT EQ 5"
123.                                            ELSE
```

```
123.                                                BEGIN   "SUB'CAT NE 5"
124.          COND13:                               IF  SUB'CAT = 3;
125.                                                  "THEN"
125.                                                    ID=WORD[9];      "GET WORD 9"
126.                                                  ELSE
127.                                                    ID=WORD[10];     "GET WORD 10"
128.          COND14:                               IF  ID = 'F';
129.                                                  "THEN"
129.                                                    BIT(WORD'16,11)=1;       "SET BIT 11"
130.                                                  ELSE
131.                                                    BIT(WORD'16,13)=1;       "SET BIT 13"
132.                                                SRN=WORD[4];    "GET WORD 4"
133.                                                PERFORM BLK'8;
134.                                                END     "SUB'CAT NE 5"
135.                                        END     "SUB'CAT NE 1"
136.                                END     "CAT EQ T"
137.
137.                            ELSE
137.                                BEGIN   "CAT NE T"
138.          COND15:               IF  CAT = 'I';
139.                                  "THEN"
139.
139.                                "WHEN CATEGORY IS I, THE CODE"
139.                                "       1. MAY CHECK VARIABLES SUB'CAT,LABEL,NON'ADDR'MSG,HOSTILE,SRN"
139.                                "                ADDRESSE,MISSION,SLOT'POINTER          "
139.                                "       2.  MAY STORE WEATHER,WEATHER'WARNING,NON'ADDR'MSG,SAM'AAA,ADDR'MSG"
139.                                "       3.  MAY RESET SLOT'POINTER"
139.
139.                                BEGIN   "CAT EQ I"
139.          COND16:               IF  SUB'CAT = 1;
140.                                  "THEN"
140.                                    BEGIN   "SUB'CAT EQ 1"
140.          COND17:                   IF  (LABEL = 1) OR (LABEL =2);
141.                                      "THEN"
141.                                        WRITEW(WEATHER);        "STORE WEATHER"
142.                                      ELSE
142.                                        BEGIN   "LABEL NE 1 OR 2"
143.          COND18:                       IF  LABEL = 3;
144.                                          "THEN"
144.                                            BEGIN   "LABEL EQ 3"
144.          COND19:                           IF  NON'ADDR'MSG = WANT;
145.                                              "THEN"
145.                                                WRITEW(NON'ADDR'MSG);   "STORE NON'ADDR'MSG"
146.                                              ELSE
147.                                                PERFORM BLK'10;
148.                                            END     "LABEL EQ 3"
149.                                          ELSE
150.                                            WRITEW(WEATHER'WARNING);        "STORE WEATHER'WARNING"
151.                                        END     "LABEL NE 1 OR 2"
152.                                    END     "SUB'CAT EQ 1"
153.                                ELSE
153.                                    BEGIN   "SUB'CAT NE 1"
```

```
154.          COND20:                          IF  SUB'CAT = 3;
155.                                               "THEN"
155.                                                  PERFORM BLK'13;
156.                                               ELSE
156.                                                  BEGIN   "SUB'CAT NE 3"
157.          COND21:                                 IF  SUB'CAT = 7;
158.                                                     "THEN"
158.                                                        BEGIN   "SUB'CAT EQ 7"
158.          COND22:                                    IF  LABEL = 1;
159.                                                        "THEN"
159.                                                           BEGIN   "LABEL EQ 1"
159.                                                           HOSTILE=WORD(6);        "GET WORD 8"
160.          COND23:                                          IF  HOSTILE = ON;
161.                                                              "THEN"
161.                                                                 WRITEW(SAM'AAA);         "STORE SAM'AAA"
162.                                                           SRN=WORD(4);    "GET WORD 4"
163.          COND24:                                          IF  SRN = DEST'TGT;
164.                                                              "THEN"
164.                                                                 WRITEW(ADDR'MSG);        "STORE ADDR'MSG"
165.                                                           END     "LABEL EQ 1"
166.                                                        ELSE
167.                                                           PERFORM BLK'13;
168.                                                     END     "SUB'CAT EQ 7"
169.                                                  ELSE
170.                                                     PERFORM BLK'10;
171.                                               END     "SUB'CAT NE 3"
172.                                            END     "SUB'CAT NE 1"
173.                                      END     "CAT EQ I"
174.
174.                                   ELSE
174.                                      BEGIN   "CAT NE I"
175.          COND25:                     IF  (CAT = 'V') OR (CAT = 'A') OR (CAT = 'M');
176.                                        "THEN"
176.
176.                                           "WHEN CATEGORY IS V, A, OR M, THE CODE"
176.                                           "       1.  MAY STORE ADDR'MSG          "
176.
176.                                        PERFORM BLK'13;
177.
177.                                     ELSE
177.                                        BEGIN   "CAT NE V, A, OR M"
178.          COND26:                       IF  CAT = 'C';
179.                                           "THEN"
179.
179.                                              "WHEN CATEGORY IS C, THE CODE    "
179.                                              "       1.  MAY CHECK SUB'CAT,LABEL,NON'ADDR'MSG,ADDRESSE,MISSION"

179.                                              "       2.  MAY STORE ADDR'MSG,NON'ADDR'MSG"
179.                                              "       3.  MAY RESET SLOT'POINTER       "
179.
179.                                           BEGIN   "CAT EQ C"
179.          COND27:                          IF (SUB'CAT = 2) AND ((LABEL = 2) OR (LABEL = 3)
179.                                              OR (LABEL = 4) OR (LABEL = 5));
180.                                           "THEN"
```

```
180.                                                                    BEGIN   "SUB'CAT EQ 2 AND LABEL EQ 2,3,4 OR 5"
180.          COND28:                                                   IF  NON'ADDR'MSG = WANT;
181.                                                                        "THEN"
181.                                                                        WRITEW(NON'ADDR'MSG);   "STORE NON'ADDR'MSG"
182.                                                                    ELSE
183.                                                                        PERFORM BLK'10;
184.                                                                    END     "SUB'CAT EQ 2 AND LABEL EQ 2,3,4 OR 5"
185.                                                                ELSE
186.                                                                    PERFORM BLK'13;
187.                                                            END     "CAT EQ C"
188.
188.                                                         ELSE
188.                                                             BEGIN   "CAT NE C"
189.          COND29:                                                   IF  CAT = 'S';
190.                                                                        "THEN"
190.                                                                        "WHEN CATEGORY IS S, THE CODE"
190.                                                                        "       1,  MAY STORE SENSOR MESSAGE"
190.
190.                                                                    BEGIN   "CAT EQ S"
190.          COND30:                                                   IF  SENSOR'MSG = WANT;
191.                                                                        "THEN"
191.                                                                        WRITEW(SENSOR'MSG);     "STORE SENSOR'MSG"
192.                                                                    END     "CAT EQ S"
193.
193.                                                             END     "CAT NE C"
194.                                                     END     "CAT NE V,A OR M"
195.                                                 END     "CAT NE I"
196.                                             END     "CAT NE T"
197.                                         END     "CAT NE P"
198.                      RETURN; "EXIT FROM ROUTINE"
199.                      END     "ACCEPT'HASH'STORE"
200.
```

STATISTIC NAME	OCCURRENCES	PERCENTAGE
CHARACTERS	18155	
LINES	361	
SYMBOLS	893	
KEY WORDS	192	21.50
AND	1	0.52
BEGIN	35	18.23
BIT	13	6.77
ELSE	26	13.54
END	35	18.23
IF	35	18.23
ITEM	32	16.67
OR	6	3.13
PROC	4	2.8
RETURN	4	2.8
TABLE	1	0.52
COMMENTS	202	22.62
DIRECTIVES	5	0.56
COMPOOL	1	20.0
EJECT	4	80.0
CONSTANTS	83	9.29
INTEGER	71	85.54
CHARACTER	12	14.46
SIGNS	373	41.77
+	1	0.27
=	71	19.3
<>	1	0.27
,	14	3.75
:	31	8.31
;	138	37.0
!	5	1.34
(	38	10.19
)	38	10.19
[	18	4.83
]	18	4.83
ABBREVIATIONS/DEFINE FORMAL PARAMETERS	32	3.58
C	2	6.25
U	30	93.75
NAMES	226	25.31
COMPOOL	1	0.44
DEFINE	15	6.64
LABEL	30	13.27
PROC	31	13.72
TABLE	1	0.44
TABLE-ITEM	17	7.52
SIMPLE-ITEM	127	56.19
DECLARATIONS	37	
SIMPLE-ITEM	31	83.78
BOOLEAN	31	100.0
TABLE	1	2.70
TABLE-ITEM	1	2.70

STATISTIC NAME	OCCURRENCES	PERCENTAGE
PROC	4	10.81
STATEMENTS	200	
SINGLE ASSIGNMENT	30	15.0
PROC CALL	27	13.50
IF	35	17.50
RETURN	4	2.0

```
STMT    LOC  R    CODE R  CODE R    LABEL     MNEM  OPERANDS
                  0028 L                      ADDR'MSG
                  0027 L                      FILTER'BIT
                  0020 L                      WORD-1
                  0026 L                      SAM'AAA
                  0029 L                      WEATHER
                  0018 L                      NOW'ADDR'M
                  002A L                      WEATHER'WA
                  0022 L                      SENSOR'MSG
 . REGION 1 *

2.          0058 H                  ACCEPT'

 . REGION 2 *

37.         0000 H                  BLK'10
 . REGION 3 *
38.         0000 H  8020  002B L            L     0,SLOT'POINT
            0002 H  F020  002C L            C     0,END'POINTE
            0004 H  7050  0030 H            JC    NE,H.C
39.         0006 H  8020  002D L            L     0,TOP'POINTE
            0008 H  9020  002B L            ST    0,SLOT'POINT
40.         000A H  70F0  0012 H            J     H.12
41.         000C H  8300  000F      H.C     LIM   0,15
            000E H  A020  002B L            A     0,SLOT'POINT
            0010 H  9020  002B L            ST    0,SLOT'POINT
43.         0012 H  70F2  0000      H.12    J     \0,2

 . REGION 4 *

45.         0014 H                  BLK'8

 . REGION 5 *
46.         0016 H  8000  0014 L            L     0,SRN
            0018 H  F000  0020 L            C     0,DEST
            001A H  7050  0024 H            JC    NE,H.24
47.         001C H  83F0  01D4 H            LIM   15,(ADDR'MSG)
            001E H  7220  0000 *            JS    2,OUTPUT
48.         0020 H  50E0  0033 L            SR    14,WORD'16
50.         0022 H  70F0  002E H            J     H.2E
51.         0024 H  8010  002F L      H.24  L     1,WORD+14
            0026 H  9010  001C L            ST    1,MSN
52.         0028 H  F010  0023 L            C     1,OWN'MSN
            002A H  7050  002E H            JC    NE,H.2E
53.         002C H  50E0  0000 L            SR    14,WORD'16
55.         002E H  83F0  01D5 H      H.2E  LIM   15,(FILTER'BIT)
            0030 H  7220  0033 *            JS    2,WRITE*
56.         0032 H  7220  0000 H            JS    2,BLK'10
58.         0034 H  8F20  0038 L      H.34  LM    2,L.38
            0036 H  70F2  0000              J     \0,2

            0014 H  9F20  0038 L            STM   2,L.38
```

```
SENT    LOCN R    CODE R  CODE R    LABEL    MNEM  OPERANDS

. REGION 6 *

6#.     0038 H                      BLK'13

. REGION 7 *
61.     003A H    8010    0004 L              L     1,WORD+3
        003C H    9010    0019 L              ST    1,ADDRESSE
62.     003E H    F010    0024 L              C     1,SELF
        0042 H    7050    0048 H              JC    NE,H.48
63.     0042 H    83F0    01D4 H              LIM   15,(ADDR'MSG)
        0044 H    7220    0000 *              JS    2,WRITEW
64.     0046 H    70F0    0054 H              J     H.54
65.     0048 H    8010    000F L    H.48      L     1,WORD+14
        004A H    9010    001D L              ST    1,MISSION
66.     004C H    F010    0025 L              C     1,OWN
        004E H    7050    0054 H              JC    NE,H.54
67.     0050 H    83F0    01D4 H              LIM   15,(ADDR'MSG)
        0052 H    7220    0000 *              JS    2,WRITEW
70.     0054 H    8F20    003C L    H.54      LM    2,L.3C
        0056 H    70F2    0000              J     \0,2

        0058 H    9F20    003C L              STM   2,L.3C

. REGION 8 *
72.     005A H    83F0    01D6 H              LIM   15,(WORD-1)
        005C H    7220    0000 *              JS    2,READR
73.     005E H    8010    0001 L              L     1,WORD
        0060 H    9010    0010 L              ST    1,XI
74.     0062 H    F310    0001      COND1     CIM   1,\1
        0064 H    7020    0068 H              JC    EQ,H.68
75.     0066 H    5010    0000 L              SB    1,WORD'16
76.     0068 H    5000    0000 L    H.68      SB    0,WORD'16
77.     006A H    8000    0011 L    COND2     L     0,CAT
        006C H    F300    5020              CIM   0,'P '
        006E H    7050    0094 H              JC    NE,H.94
78.     0070 H    8010    0012 L    COND3     L     1,SUB'CAT
        0072 H    F310    0001              CIM   1,\1
        0074 H    7050    0082 H              JC    NE,H.82
79.     0076 H    5090    0000 L              SB    9,WORD'16
80.     0078 H    8000    0013 L    COND4     L     0,FLT
        007A H    F000    001E L              C     0,LOR
        007C H    7050    0080 H              JC    NE,H.80
81.     007E H    5FF0    0000 L              SB    15,WORD'16
83.     0080 H    70FF    008C H    H.80      J     H.8C
84.     0082 H                      H.82
        0082 H    F310    0003      COND5     CIM   1,\3
        0084 H    7050    008A H              JC    NE,H.8A
85.     0086 H    5010    0000 L              SB    1#,WORD'16
86.     0088 H    70F0    008C H              J     H.8C
87.     008A H    5080    0000 L    H.8A      SB    11,WORD'16
89.     008C H    8000    0002 L    H.8C      L     0,WORD+1
```

```
STMT   LOCN R   CODE R   CODE R   LABEL     MNEM   OPERANDS
       0092 H   9000     0014 L             ST     0,SRN
90.    0094 H   7220     0014 H             JS     2,BLK'8
92.    0096 H   70F0     01D0 H             J      H.1D0
93.    0098 H                     H.94
       0098 H   8000     0011 L   COND6     L      0,CAT
       009A H   F300     5420               CIM    0,'T
       009C H   7050     010E H             JC     NE,H.10E
94.    009E H   8310     0001     COND7     LIM    1,\1
       00A0 H   F010     0012 L             C      1,SUB'CAT
       00A2 H   7050     00B6 H             JC     NE,H.B6
95.    00A4 H   8220     0009 L             L      2,WORD+8
       00A6 H   9020     0015 L             ST     2,ID
96.    00A8 H   F320     0006     COND8     CIM    2,7
       00AA H   7050     00AC H             JC     NE,H.AC
97.    00AC H   5090     0000 L             SB     9,WORD'16
98.    00AE H   70F0     00AE H             J      H.AE
99.    00B0 H   50C0     0000 L   H.AC      SB     12,WORD'16
100.   00B2 H   8000     0004 L   H.AE      L      0,WORD+3
       00B4 H   9000     0014 L             ST     0,SRN
101.   00B6 H   7220     0014 H             JS     2,BLK'8
103.   00B8 H   70F0     01DC H             J      H.1DC
104.   00BA H                     H.B6
       00BA H   8000     0012 L   COND9     L      0,SUB'CAT
       00BC H   F300     0005               CIM    0,\5
       00BE H   7050     00EA H             JC     NE,H.EA
105.   00C0 H   8010     0004 L             L      1,WORD+9
       00C2 H   9010     0015 L             ST     1,ID
106.   00C4 H   F310     0006     COND10    CIM    1,7
       00C6 H   7050     00CE H             JC     NE,H.CE
107.   00C8 H   5080     0000 L             SB     11,WORD'16
108.   00CA H   8020     0004 L             L      0,WORD+3
       00CC H   9000     0014 L             ST     0,SRN
109.   00CE H   7220     0014 H             JS     2,BLK'8
111.   00D0 H   70F0     00E8 H             J      H.E8
112.   00D2 H                     H.CE
       00D2 H   8000     0016 L   COND11    L      0,FIRST'LEVE
       00D4 H   F300     0001               CIM    0,\1
       00D6 H   7050     00E6 H             JC     NE,H.E6
113.   00D8 H   8000     0017 L   COND12    L      0,SECOND'LEV
       00DA H   F300     0003               CIM    0,\3
       00DC H   7050     00E4 H             JC     NE,H.E4
114.   00DE H   83F0     01D7 H             LIM    15,(GAA'AAA)
       00E0 H   7220     0000 *             JS     2,OUTPUT
115.   00E2 H   8000     0004 L             L      0,WORD+3
       00E4 H   9000     0014 L             ST     0,SRN
116.   00E6 H   7220     0014 H             JS     2,BLK'8
117.   00E8 H   70F0     00E8 H   H.E4      J      H.E8
120.   00EA H   7220     0000 H   H.E6      JS     2,BLK'10
123.   00EC H   70F0     01DC H   H.E8      J      H.1DC
       00EA H                     H.EA
. REGION 9 .
124.   00EA H   8000     0012 L   COND13    L      0,SUB'CAT
```

```
STMT    LOCN R    CODE R  CODE R    LABEL     MNEM   OPERANDS
        00EC H    F300    0003              CIM    0,\3
        00EE H    7A50    00F6 H            JC     NE,H.F6
125.    00F0 H    8000    0009 L            L      0,WORD+8
        00F2 H    9000    0015 L            ST     0,ID
126.    00F4 H    70F0    00FA H            J      H.FA
127.    00F6 H    8000    000A L    H.F6    L      0,WORD+9
        00F8 H    9000    0015 L            ST     0,ID
128.    00FA H                      H.FA
        00FA H    8000    0015 L    COND14  L      0,ID
        00FC H    F300    0046              CIM    0,70
        00FE H    7050    0104 H            JC     NE,H.104
129.    0100 H    5080    0000 L            SB     11,WORD'16
130.    0102 H    70F0    0106 H            J      H.106
131.    0104 H    5080    0000 L    H.104   SB     13,WORD'16
132.    0106 H    8000    0004 L    H.106   L      0,WORD+3
        0108 H    9000    0014 L            ST     0,SRN
133.    010A H    7220    0014 H            JS     2,BLK'8
134.    010C H                      H.10C
. REGION 10*
135.    010C H                      H.10C
. REGION 11*
137.    010C H    70F0    01D0 H            J      H.1D0
        010E H                      H.10E
. REGION 12*
138.    010E H    8000    0011 L    COND15  L      0,CAT
        0110 H    F300    4920              CIM    0,'I '
        0112 H    7050    017C H            JC     NE,H.17C
139.    0114 H    8310    0001      COND16  LIM    1,\1
        0116 H    F010    0012 L            C      1,SUB'CAT
        0118 H    7050    0144 H            JC     NE,H.144
140.    011A H    8010    001A L    COND17  L      1,LABEL
        011C H    F310    0001              CIM    1,\1
        011E H    7020    0124 H            JC     EQ,H.124
        0120 H    F310    0002              CIM    1,\2
        0122 H    7050    012A H            JC     NE,H.12A
141.    0124 H    83F0    01D8 H    H.124   LIM    15,(WEATHER)
        0126 H    7220    0000 *            JS     2,WRITEW
142.    0128 H    70F0    0142 H            J      H.142
143.    012A H                      H.12A
        012A H    F310    0003      COND18  CIM    1,\3
        012C H    7050    013E H            JC     NE,H.13E
144.    012E H    8010    001F L    COND19  L      1,NON'ADDR'M
        0130 H    F010    001F L            C      1,WANT
        0132 H    7050    013A H            JC     NE,H.13A
145.    0134 H    83F0    01D9 H            LIM    15,(NON'ADDR'M)
        0136 H    7220    0000 *            JS     2,WRITEW
146.    0138 H    70F6    013C H            J      H.13C
147.    013A H    7220    0000 H    H.13A   JS     2,BLK'10
149.    013C H    70F0    0142 H    H.13C   J      H.142
150.    013E H    83F0    01DA H    H.13E   LIM    15,(WEATHER'WA)
        0140 H    7220    0000 *            JS     2,WRITEW
153.    0142 H    70F6    017A H    H.142   J      H.17A
```

```
STMT   LOCN R    CODE F  CODE R   LABEL     MNEM  OPERANDS
154.   0144 H                     H.144
       0144 H    8000    0012 L   COND20    L     0,SUB'CAT
       0146 H    F320    0003             CIM   0,\3
       0148 H    7050    014E H           JC    NE,H.14E
155.   014A H    7220    0038 H           JS    2,BLK'13
156.   014C H    70F0    017A H           J     H.17A
157.   014E H                     H.14E
       014E H    8000    0012 L   COND21    L     0,SUB'CAT
       0150 H    F320    0007             CIM   0,\7
       0152 H    7050    0178 H           JC    NE,H.178
158.   0154 H    8310    0001     COND22    LIM   1,\1
       0156 H    F410    001A L           C     1,LABEL
       0158 H    7050    0174 H           JC    NE,H.174
159.   015A H    8220    0008 L           L     2,WORD+7
       015C H    9020    001B L           ST    2,HOSTILE
160.   015E H    F320    0001     COND23    CIM   2,\1
       0160 H    7050    0166 H           JC    NE,H.166
161.   0162 H    83F0    01D7 H           LIM   15,(SAM'AAA)
       0164 H    7220    0000 *           JS    2,WRITEW
162.   0166 H    8010    0004 L   H.166     L     1,WORD+3
       0168 H    9010    0014 L           ST    1,SRV
163.   016A H    F410    0021 L   COND24    C     1,DEST'TGT
       016C H    7050    0172 H           JC    NE,H.172
164.   016E H    83F0    01D4 H           LIM   15,(ADDR'MSG)
       0170 H    7220    0000 *           JS    2,WRITEW
166.   0172 H    70F0    017B H   H.172     J     H.17B
167.   0174 H    7220    0038 H   H.174     JS    2,BLK'13
169.   0176 H    70F0    017A H   H.17B     J     H.17A
170.   0178 H    7220    0038 H   H.178     JS    2,BLK'10
174.   017A H    70F0    01D4 H   H.17A     J     H.1D0
175.   017C H                     H.17C
       017C H    8010    0011 L   COND25    L     1,CAT
       017E H    F310    5620             CIM   1,'V  '
       0180 H    7020    018A H           JC    EQ,H.18A
       0182 H    F310    4120             CIM   1,'A  '
       0184 H    7020    018A H           JC    EQ,H.18A
       0186 H    F310    4D20             CIM   1,'M  '
       0188 H    7050    018E H           JC    NE,H.18E
176.   018A H    7220    0038 H   H.18A     JS    2,BLK'13
177.   018C H    70F0    01D0 H           J     H.1D0
178.   018E H                     H.18E
       018E H    8000    0011 L   COND26    L     0,CAT
       0192 H    F320    4320             CIM   0,'C  '
       0192 H    7050    01C0 H           JC    NE,H.1C0
179.   0194 H    8310    0002     COND27    LIM   1,\2
       0196 H    F410    0012 L           C     1,SUB'CAT
       0198 H    7050    019C H           JC    NE,H.19C
       019A H    8020    001A L           L     2,LABEL
       019C H    F320    0002             CIM   2,\2
       019E H    7020    01AC H           JC    EQ,H.1AC
       01A0 H    F320    0003             CIM   2,\3
       01A2 H    7020    01AC H           JC    EQ,H.1AC
```

```
STMT    LOCN R    CODE R  CODE R    LABEL     MNEM    OPERANDS
        01A4 H    F320    0004                CIM     2,\4
        01A6 H    7020    01AC H              JC      EQ,H.1AC
        01A8 H    F320    0005                CIM     2,\5
        01AA H    7050    01BC H              JC      NE,H.1BC
180.    01AC H                      H.1AC
        01AC H    8010    0018 L    COND28    L       1,NON'ADDR'M
        01AE H    F010    001F L              C       1,WANT
        01B0 H    7050    01B8 H              JC      NE,H.1B8
181.    01B2 H    83E0    01D9 H              LIM     15,(NON'ADDR'M)
        01B4 H    7220    0000 *              JS      2,WRITEW
182.    01B6 H    70E0    01BA H              J       H.1BA
183.    01B8 H    7220    0000 H    H.1B8     JS      2,BLK'10
185.    01BA H    70E0    01BE H    H.1BA     J       H.1BE
186.    01BC H    7220    0038 H    H.1BC     JS      2,BLK'13
188.    01BE H    70E0    01D0 H    H.1BE     J       H.1D0
189.    01C0 H                      H.1C0
        01C0 H    8000    0011 L    COND29    L       0,CAT
        01C2 H    E300    5320                CIM     0,'S  '
        01C4 H    7050    01D0 H              JC      NE,H.1D0
190.    01C6 H    8010    0022 L    COND30    L       1,SENSOR'MSG
        01C8 H    F010    001F L              C       1,WANT
        01CA H    7050    01D0 H              JC      NE,H.1D0
191.    01CC H    83E0    01DB H              LIM     15,(SENSOR'MSG)
        01CE H    7220    0000 *              JS      2,WRITEW
196.    01D0 H                      H.1D0
. REGION 13*
197.    01D0 H                      H.1D0
. REGION 14*
199.    01D0 H    8F20    0030 L    H.1D0     LM      2,L.30
        01D2 H    70E2    0000                J       \0,2

        0258 H    9F20    0030 L              STM     2,L.30
. REGION 15*
200.                                          END
```

NAME	CLAS	SCOPE	LOC	TY	F8	SIZ	PERM/DSIZ	DEF	- - SET(*)/USED - -								
ACCEPT'HAS	PROC	*GLOBAL*	0058H				7	2:									
ADDR'MSG	ITEM	ACCEPT'HAS	0028L	U	0	16	RESERV	30:	47	63	67	164					
ADDRESSF	ITEM	ACCEPT'HAS	0019L	U	0	16	RESERV	15:	61*	62							
ATA:	PROC	*GLOBAL*	0000	F	0	32	0	2:									
BLK'10	PROC	ACCEPT'HAS	0000H				2	37:	56	56	120	120	147	147	170	170	183
									183								
BLK'13	PROC	ACCEPT'HAS	0038H				4	60:	155	155	167	167	176	176	186	186	
BLK'8	PROC	ACCEPT'HAS	0014H				5	45:	90	90	101	101	109	109	116	116	133
									133								
CAT	ITEM	ACCEPT'HAS	0011L	C	0	2	RESERV	7:	77	93	138	175	175	175	178	189	
COND1	LABL	ACCEPT'HAS	0062H					74:									
COND10	LABL	ACCEPT'HAS	00C0H					106:									
COND11	LABL	ACCEPT'HAS	00CEH					112:									
COND12	LABL	ACCEPT'HAS	00D4H					113:									
COND13	LABL	ACCEPT'HAS	00EAH					124:									
COND14	LABL	ACCEPT'HAS	00FAH					128:									
COND15	LABL	ACCEPT'HAS	010EH					138:									
COND16	LABL	ACCEPT'HAS	0114H					139:									
COND17	LABL	ACCEPT'HAS	011AH					140:									
COND18	LABL	ACCEPT'HAS	012AH					143:									
COND19	LABL	ACCEPT'HAS	012EH					144:									
COND2	LABL	ACCEPT'HAS	006AH					77:									
COND20	LABL	ACCEPT'HAS	0144H					154:									
COND21	LABL	ACCEPT'HAS	014EH					157:									
COND22	LABL	ACCEPT'HAS	0154H					158:									
COND23	LABL	ACCEPT'HAS	015EH					160:									
COND24	LABL	ACCEPT'HAS	016AH					163:									
COND25	LABL	ACCEPT'HAS	017CH					175:									
COND26	LABL	ACCEPT'HAS	018EH					178:									
COND27	LABL	ACCEPT'HAS	0194H					179:									
COND28	LABL	ACCEPT'HAS	01ACH					180:									
COND29	LABL	ACCEPT'HAS	01CEH					189:									
COND3	LABL	ACCEPT'HAS	0070H					78:									
COND30	LABL	ACCEPT'HAS	01C6H					190:									
COND4	LABL	ACCEPT'HAS	0078H					80:									
COND5	LABL	ACCEPT'HAS	0082H					84:									
COND6	LABL	ACCEPT'HAS	0094H					93:									
COND7	LABL	ACCEPT'HAS	009AH					94:									
COND8	LABL	ACCEPT'HAS	00A4H					96:									
COND9	LABL	ACCEPT'HAS	00B6H					104:									
COS	PROC	*GLOBAL*	0000	F	0	32	0	2:									
DATA	LABL	ACCEPT'HAS	0001L	U	0	16	RESERV	4:									
DEST	ITEM	ACCEPT'HAS	0020L	U	0	16	RESERV	22:	46								
DEST'TGT	ITEM	ACCEPT'HAS	0021L	U	0	16	RESERV	23:	163								
END'POINTE	ITEM	ACCEPT'HAS	002CL	U	0	16	RESERV	34:	38								
FILTER'BIT	ITEM	ACCEPT'HAS	0027L	U	0	16	RESERV	29:	55								
FIRST'LEVE	ITEM	ACCEPT'HAS	0016L	U	0	16	RESERV	12:	112								
FLT	ITEM	ACCEPT'HAS	0013L	U	0	16	RESERV	9:	80								
HOSTILE	ITEM	ACCEPT'HAS	0018L	U	0	16	RESERV	17:	159*	160							
ID	ITEM	ACCEPT'HAS	0015L	U	0	16	RESERV	11:	95*	96	105*	106	125*	127*	128		
JTIDS	BLOC	*GLOBAL*	0000*			0	EXTRNL	2:									
LABEL	ITEM	ACCEPT'HAS	0001AL	U	0	16	RESERV	16:	140	140	143	158	179	179	179	179	

NAME	CLAS	SCOPE	LOC	TY	FB	SIZ	PERM/DSIZ	DEF	- - SET(*)/USED - -								
LOR	ITEM	ACCEPT'HAS	0010L	U	0	16	RESERV	20:	80								
MISSION	ITEM	ACCEPT'HAS	0010L	U	0	16	RESERV	19:	65*	66							
MSN	ITEM	ACCEPT'HAS	0010L	U	0	16	RESERV	18:	51*	52							
NON'ADDR'M	ITEM	ACCEPT'HAS	0018L	U	0	16	RESERV	14:	144	145	180	181					
ON	DEIN	*GLOBAL*						2:	160								
OUTPUT	PROC	*GLOBAL*	0000*				0	2:	47	114							
OWN	ITEM	ACCEPT'HAS	0025L	U	0	16	RESERV	27:	66								
OWN'MSN	ITEM	ACCEPT'HAS	0023L	U	0	16	RESERV	25:	52								
PERFORM	DEIN	*GLOBAL*						2:	56	90	101	109	116	120	133	147	155
									167	170	176	183	186				
READR	PROC	*GLOBAL*	0000*				0	2:	72								
SAM'AAA	ITEM	ACCEPT'HAS	0020L	U	0	16	RESERV	26:	114	161							
SECOND'LEV	ITEM	ACCEPT'HAS	0017L	U	0	16	RESERV	13:	113								
SELF	ITEM	ACCEPT'HAS	0024L	U	0	16	RESERV	26:	62								
SENSOR'MSG	ITEM	ACCEPT'HAS	0022L	U	0	16	RESERV	24:	190	191							
SIN	PROC	*GLOBAL*	0000	F	0	32	0	2:									
SLOT'POINT	ITEM	ACCEPT'HAS	0026L	U	0	16	RESERV	33:	38	39*	41	41*					
SQRT	PROC	*GLOBAL*	0000	F	0	32	0	2:									
SRN	ITEM	ACCEPT'HAS	0014L	U	0	16	RESERV	10:	46	89*	100*	108*	115*	132*	162*	163	
SUB'CAT	ITEM	ACCEPT'HAS	0012L	U	0	16	RESERV	8:	78	84	94	104	124	139	154	157	179
TAN	PROC	*GLOBAL*	0000	F	0	32	0	2:									
TOP'POINTE	ITEM	ACCEPT'HAS	0028L	U	0	16	RESERV	35:	39								
WANT	ITEM	ACCEPT'HAS	0016L	U	0	16	RESERV	21:	144	180	190						
WEATHER	ITEM	ACCEPT'HAS	0029L	U	0	16	RESERV	31:	141								
WEATHER'WA	ITEM	ACCEPT'HAS	0027L	U	0	16	RESERV	32:	150								
WORD	ITEM	DATA	0001L	C	0	2	RESERV	5:	51	61	65	72	73	89	95	100	105
									108	115	125	127	132	159	162		
WORD'16	ITEM	ACCEPT'HAS	0000L	U	0	16	RESERV	3:	48*	53*	75*	76*	79*	81*	85*	87*	97*
									99*	107*	129*	131*					
WRITEN	PROC	*GLOBAL*	0000				0	2:	55	63	67	141	145	150	161	164	181
									191								
XI	ITEM	ACCEPT'HAS	0010L	U	0	16	RESERV	6:	73*	74							

PROGRAM SUMMARY

DATA/VARIABLES 0000 - 003F
INSTRUCTIONS/CONSTANTS 8000 - 81DB
EXTERNS: WRITEN READR OUTPUT JTIDS
INTERNS: ACCEPT
FILES REFERENCED:
JTIDS.CMP 9/14/76 9:36 CMP:JTIDS

361 LINES 92 MESSAGES: 92 INFORMATION
CPU TIME 11.827 SEC

LPTSPL Version 6(347) Running on LPTM11
START User AUROA&OFF [3232,1376] Job JTIDS Seq. 521 Date 14-Sep-76 10:03:39 Monitor AFAL 6#2.15 SYSTEM *START*
Request created: 14-Sep-76 10:04:11
File: DSKD1:JTIDS.LST[3232,1376] Created: 14-Sep-76 09:35:00 Printed: 14-Sep-76 10:04:02
QUEUE Switches: /PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:372 /FORMS:NORMAL
File will be deleted after printing

```
JOVIAL  V.Ø62876  9/14/76 9:35  MODULE:JTIDS.JTS     PAGE 1

JTIDS,JTIDS=JTIDS.JTS,JTIDS/HBC/MAC/NOIN

    1.          COMPOOL JTIDS;
    2.                  BEGIN
    2.                  DEFINE  PERFORM " ";
    3.                  DEFINE  ON "1";
    4.                  PROC    READR(INPUT);
    5.                          BEGIN
    6.                          ITEM     INPUT U;;
    7.                          END
    8.                  PROC    WRITEW(STORE);
    9.                          BEGIN
    9.                          ITEM     STORE   U;
   10.                          END
   11.                  PROC    OUTPUT(SOME);
   12.                          BEGIN
   12.                          ITEM     SOME    U;
   13.                          END
   14.                  PROC    ATAN(VAL)       F;
   15.                          BEGIN
   15.                          ITEM     VAL     F;
   16.                          END
   17.                  PROC    SIN(VAL1)       F;
   18.                          BEGIN
   18.                          ITEM     VAL1    F;
   19.                          END
   20.                  PROC    COS(VAL2)       F;
   21.                          BEGIN
   21.                          ITEM     VAL2    F;
   22.                          END
   23.                  PROC    SQRT(VAL3)      F;
   24.                          BEGIN
   24.                          ITEM     VAL3    F;
   25.                          END
   26.                  PROC    TAN(VAL4)       F;
   27.                          BEGIN
   27.                          ITEM     VAL4    F;
   28.                          END
   28.                  END

JOVIAL  V.Ø62876  9/14/76 9:35  MODULE:JTIDS.JTS     PAGE 2
STMT    LOCN R      CODE R  CODE R     LABEL    MNEM  OPERANDS

 . REGION 1 *
30.                                                 END

PROGRAM SUMMARY
DATA/VARIABLES              ØØØØ - FFFF
INSTRUCTIONS/CONSTANTS      8ØØØ - 7FFF
EXTERNS:  NONE
INTERNS: JTIDS
FILES REFERENCED:  NONE

37 LINES  Ø MESSAGES
CPU TIME      1.657 SEC
```